

Aplicación web, presentaciones dinámicas

‘Cool Presentations’

Tutor: *Carlos Álvarez*

Alumno: *Esteve Oria*

Especialidad: *Tecnologías de la información*



Universitat Politècnica de Catalunya (UPC)
Facultat de Informàtica de Barcelona (FIB)

2018

“Para comunicarnos efectivamente, debemos darnos cuenta de que todos somos diferentes en la forma en que percibimos el mundo y usar ese conocimiento como guía para comunicarnos con otros.”

-Tony Robbins-

“El éxito de tu presentación vendrá dado no por el conocimiento que transmitas sino por el que reciba el público.”

–Lily Walters-

Resumen

Este proyecto tiene la finalidad de desarrollar una aplicación web que ayude a mejorar el seguimiento de presentaciones, además, pretende mejorar el feedback entre presentador y audiencia para cualquier tipo de presentación en la que se usen diapositivas como soporte. Para ello, la aplicación añadirá diversas herramientas o funcionalidades a una presentación con transparencias.

Gracias a las funcionalidades que la aplicación añade a las presentaciones, y al extendido uso de dispositivos móviles y portables, se pretende lograr una comunicación más eficiente, dejando atrás las tradicionales presentaciones en las que el único que actúa es el orador.

Resum

Aquest projecte té la finalitat de desenvolupar una aplicació web que ajudi a millorar el seguiment de presentacions, a més, pretén millorar també el feedback entre presentador i audiència per a qualsevol tipus de presentació en què es facin servir diapositives com a suport. Per a això, l'aplicació afegirà diverses eines o funcionalitats a una presentació amb transparències.

Gràcies a les funcionalitats que l'aplicació afegeix a les presentacions, i l'estès ús de dispositius mòbils i portables, es pretén aconseguir una comunicació més eficient, deixant enrere les tradicionals presentacions en les que l'únic que actua és l'orador.

Abstract

This project has the purpose of developing a web application that helps to improve the tracking of presentations, also aims to improve the feedback between

presenter and audience for any type of presentation in which slides are used as support. For this, the application will add various tools or functionalities to a presentation with slides.

Thanks to the functionalities that the application adds to the presentations, and the widespread use of mobile and portable devices, it is intended to achieve a more efficient communication, leaving behind the traditional presentations in which the only one acting is the speaker.

Contenidos

Resumen	1
Contenidos	3
1 Introducción	6
1.1 Aplicación web	7
1.2 Áreas de interés	7
1.3 Actores implicados	7
2 Glosario	9
3 Estado del arte	11
4 Formulación del problema	13
4.1 Objetivos	13
4.2 Alcance	13
5 Metodología y rigor	15
5.1 Herramientas de desarrollo	15
5.2 Seguimiento	16
5.3 Validación de resultados	17
5.4 Obstáculos y riesgos	18
5.4.1 Bugs	18
5.4.2 Bajo presupuesto	18
6 Planificación temporal	20
6.1 Descripción de las tareas	20
6.1.1 Estudio viabilidad	20
6.1.2 Estudio contexto, planificación temporal y costes	21
6.1.3 Preparación entorno	21
6.1.4 Desarrollo del proyecto	22
6.1.5 Verificación y finalización	22
6.2 Planificación y tiempo estimado	23
6.3 Diagrama de Gantt	24
6.4 Desviaciones y plan de acción	24

6.5 Modificaciones planificación temporal, desviaciones	25
7 Implementación	28
7.1 Capa presentación	28
7.1.1 Librerías utilizadas	33
7.2 Capa de datos	34
7.2.1 Estructura base de datos	37
7.3 Funciones / Algoritmos principales	40
7.3.1 Añadir dudas	40
7.3.2 “Near Real Time” resultado encuestas, AJAX vs WebSockets	43
7.3.3 Uso de plantillas html y javascript	44
7.3.4 Comprobación de usuario y contraseña	47
7.3.5 Obtener presentaciones a partir de una condición	51
7.3.6 Index.php protección contra ataques	53
8 Servidor y calidad de servicio	56
8.1 Droplet, DigitalOcean	56
8.1.1 Apache	58
8.1.2 Monitorización	59
8.2 CDN, CloudFlare	66
8.3 CA, Let’s Encrypt - CertBot	66
8.4 Dominio, NameCheap	68
9 Gestión económica y sostenibilidad	70
9.1 Autoevaluación competencia sostenibilidad	70
9.2 Dimensión económica - presupuesto	71
9.2.1 Costes directos	71
9.2.2 Costes indirectos	72
9.2.3 Costes desviaciones e imprevistos “previstos”	73
9.2.4 Control de costes	73
9.2.5 Costes totales	74
9.2.6 Reflexión sostenibilidad económica	75
9.3 Dimensión ambiental	75
9.4 Dimensión social	76

10 Integración de conocimientos	78
11 Identificación de leyes y regulaciones	80
12 Conclusiones y reflexiones finales	84
12.1 Mejoras de futuro	85
13 Bibliografía	87
14 Índice de figuras	91
15 Índice de tablas	94
16 Anexo I: Manual de usuario	95

1 Introducción

Actualmente vivimos en una época llena de distracciones, es por ello que el tiempo, atención y opinión de las personas son muy preciados en la sociedad moderna. Grandes entidades recompensan económicamente a un público que les da su atención y tiempo. De modo que las opiniones de sus clientes son su principal factor diferenciador y marcan el camino que deben seguir. Por esta razón, las encuestas de satisfacción cada vez son más valoradas y las recompensas por completarlas mayores. En definitiva, la retroalimentación, o más comúnmente conocido como “feedback”, es esencial para conseguir el éxito en casi cualquier ámbito.

Como hemos comentado, captar la atención de la gente para que pueda darnos su opinión es esencial, pero no es una tarea fácil. Hoy en día estamos conectados 24 horas al día los 7 días de la semana, y conseguir que alguien desconecte para darte su atención es complicado. Incluso en el trato directo cara a cara, cuesta conseguir una atención plena. Para conseguir dicha atención debemos tener distraído a nuestro público con el tema que queremos tratar. De aquí surge la iniciativa para llevar a cabo este proyecto, cuya finalidad es captar con mayor facilidad la atención de la gente y mejorar el “feedback” en cualquier interacción / presentación / reunión / clase / conferencia con transparencias.

Con este proyecto pretendo poner en práctica muchos de los conocimientos adquiridos a lo largo de mi carrera universitaria y desarrollar un servicio online que permita modernizar, o completar, las clásicas presentaciones con transparencias facilitando el seguimiento y la retroalimentación entre presentador y audiencia.

Este escrito tiene la finalidad de darle al lector las nociones necesarias para poder comprender el proyecto, el estado en el que se encuentra y cómo se ha realizado.

1.1 Aplicación web

Este proyecto se fundamenta en la creación de una aplicación web que ayudará a mejorar el seguimiento y feedback en presentaciones con transparencias. Las aplicaciones web permiten a los usuarios acceder a herramientas accediendo a un servidor mediante un navegador web. Son ampliamente extendidas debido a la practicidad del navegador web como cliente ligero, además, no depende del sistema operativo, esto hace que sea fácilmente actualizable y mantenible.

1.2 Áreas de interés

El uso de presentaciones con transparencias es muy extendido. Podemos encontrar ejemplos claros en multitudes de entornos ya que es una de las mejores formas de comunicar a un público una idea. Claros ejemplos son profesores, tanto de escuela como universidad y conferenciantes. En el ámbito corporativo su uso es ampliamente reconocido y recomendado.

1.3 Actores implicados

A continuación se presentan los actores implicados en el desarrollo del proyecto:

Desarrollador y director: Son las personas encargadas del desarrollo del proyecto. El director se encarga de la supervisión.

Usuarios: Son aquellas personas que suben una presentación a la web para añadirle funcionalidades. Usuarios directo también son la audiencia que sigue la presentación mediante su dispositivo móvil.

Otros beneficiarios: Corresponde a aquella audiencia, que sin estar haciendo uso de la web en sí se ven beneficiados por estar atendiendo una presentación con funcionalidades añadidas por el proyecto.

2 Glosario

Antes de empezar con el proyecto necesitamos conocer algunos términos que aparecerán varias veces a lo largo del documento.

feedback: capacidad de un emisor para recoger reacciones de los receptores y modificar su mensaje, de acuerdo con lo recogido.

droplet: es el nombre que la empresa DigitalOcean da a sus máquinas virtuales.

hosting: o alojamiento web, es el espacio donde se aloja un sitio web para que sea accesible desde internet.

funcionalidades añadidas: hace referencia a las funcionalidades que el proyecto pretende añadir a una presentación clásica con diapositivas, por ejemplo, la posibilidad de añadir encuestas sobre las diapositivas.

arrastrable: hace referencia al gesto gráfico de arrastrar y soltar sobre la pantalla. También conocido como “drag and drop”.

responsive: técnica de diseño que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles.

proveedor cloud: hace referencia a las empresas que ofrecen servicios de computación en la nube.

hashear: hace referencia al hecho de usar la función hash, es decir, transformar cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija.

3 Estado del arte

En las siguientes líneas se darán a conocer proyectos con objetivos y características similares que hayan sido desarrollados con las últimas tecnologías, probados en la industria y acogidos por diferentes entidades. Tener en cuenta éstas investigaciones ayudará a poder desarrollar un proyecto puntero tecnológicamente y con características diferenciadoras. Además, la revisión de herramientas parecidas ayudará a aclarar el objetivo y alcance del proyecto.

La gran parte de herramientas de presentaciones que encontramos por la red nos ayudan a crear transparencias desde cero. Algunas con diseños más sencillos y otras con diseños dinámicos, sin embargo, el objetivo de este proyecto no tiene nada que ver con estas herramientas. Por este motivo, sólo se tendrán en cuenta las herramientas que además de crear transparencias ayuden a mejorar el feedback o tengan características comunes con este proyecto.

Empezaremos hablando de una de las herramientas para crear presentaciones más conocida, Powerpoint de Microsoft [1]. Ésta dispone de un complemento llamado “Cuestionario de respuesta múltiple” que permite crear encuestas en una transparencia. No obstante, el objetivo de estas encuestas difiere mucho del proyecto. En este caso las encuestas únicamente sirven para que la audiencia pueda poner a prueba sus conocimientos aprendidos de forma individual. En ningún caso ayudan a mejorar el feedback entre presentador y audiencia o a mejorar el seguimiento de la presentación.

Otra herramienta, la cual funciona como servicio web igual que el proyecto, es Kahoot [2]. Ésta herramienta se define como una plataforma para crear cuestionarios de evaluación, básicamente tests que pueden ser replicados a dispositivos móviles. Su

finalidad es educativa, y es usada principalmente por profesores, puesto que facilita la evaluación y aprendizaje. Aunque presenta funcionalidades tecnológicamente parecidas a algunas planteadas en el proyecto, como la creación de encuestas personalizadas, éstas no tienen el mismo objetivo ni finalidad y es por ello que su planteamiento es completamente distinto.

“Poll everywhere” [3] es un complemento de Google Presentaciones. Con este complemento podemos añadir encuestas y ser contestadas por la audiencia con sus dispositivos móviles en tiempo real. “Poll everywhere” representa una de las funcionalidades de nuestro proyecto, y nos será de gran ayuda para orientarnos. Sin embargo, este complemento tiene muchas limitaciones. La principal es que solo se pueden añadir encuestas en presentaciones de Google y la segunda que hay otras muchas funcionalidades que no contempla, como por ejemplo, la anotación de dudas por parte de la audiencia, o el acceso a la presentación mediante ubicación y rango horario, entre otras muchas.

Finalmente, a partir del “paper” “Reflexiones sobre el uso de transparencias en clase” [4] de David López, profesor de la UPC, podemos afirmar varias conclusiones que nos servirán para corroborar la utilidad de nuestro proyecto. En este “paper” se defiende que un buen uso de presentaciones con transparencias puede mejorar y facilitar el aprendizaje. Además, el uso de las transparencias obliga al orador a prepararse un guión y esto repercute en una comunicación más eficiente.

4 Formulación del problema

Esta sección pretende describir y definir el objetivo del proyecto. Primeramente empezará definiendo el objetivo principal. Seguidamente, identificará el alcance y los aspectos que cubrirá el proyecto.

4.1 Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación web que ayude a mejorar el seguimiento de presentaciones, además, pretende mejorar también el feedback entre presentador y audiencia para cualquier tipo de presentación en la que se usen transparencias como soporte. Para ello, el webservice añadirá diversas funcionalidades a una presentación con transparencias. Para simplificar, los usuarios suben sus propias transparencias y la aplicación web permite incorporarle distintas utilidades según sus necesidades, con el fin de mejorar el seguimiento y feedback de distintas maneras.

4.2 Alcance

En cuanto a las funcionalidades que el proyecto pretende añadir a una presentación dependerá de los tiempos de entrega y de las posibles desviaciones. Sin embargo, hay un mínimo que garantizará:

- Acceso a una presentación mediante ubicación.
- Acceso a una presentación mediante contraseña.
- Acceso a una presentación mediante rango horario.
- Acceso a una presentación mediante la combinación de las tres funcionalidades anteriores.
- Seguimiento de la presentación mediante dispositivos móviles.
- Posibilidad de que la presentación sea descargable.

- Encuestas durante la presentación, resultados y participación en tiempo real.
- Anotar dudas sobre las transparencias con los dispositivos móviles.
- Recordatorio de dudas al final de la presentación.
- Ranking de respuestas correctas.
- Encuesta de satisfacción sobre la presentación al final de la presentación.

Esta lista de funcionalidades marcan los objetivos del proyecto.

5 Metodología y rigor

Para el desarrollo del proyecto usaremos metodologías ágiles que permiten desarrollos flexibles y rápidos, en concreto combinaremos Kanban [5] con Scrum [6]. Mencionar que, aunque ambas metodologías son diferentes no son excluyentes entre ellas. Aunque las dos están orientadas a trabajos grupales, se pueden extraer muchos conceptos para agilizar el trabajo de una sola persona. En los siguientes apartados se describen las herramientas que se utilizarán para desarrollar y para conseguir una metodología ágil y eficaz.

5.1 Herramientas de desarrollo

En cuanto a las herramientas Software necesarias para desarrollar el proyecto, únicamente necesitamos un SO y un editor de código. En concreto se han seleccionado Windows 10 [7] y Visual Code [8]. Debido al tipo de proyecto, las herramientas de desarrollo corresponden más a lenguajes de programación para cada una de las partes del proyecto.

Para la capa de presentación, o más conocida como “front-end” utilizaremos los siguientes lenguajes: HTML5, CSS3, Bootstrap, Javascript. La elección de estos lenguajes es sencilla, ya que son los más usados en la mayoría de webs de hoy en día y los que más comunidad tienen para dar soporte. Para las peticiones y respuestas de la Rest API [9] utilizaremos JSON ya que junto a XML son los dos estándares más aceptados hoy en día.

Finalmente, para la capa de acceso a datos, o más conocida como “back-end” utilizaremos PHP. El motivo de esta elección es que es uno de los lenguajes más extendidos para este uso y presenta una mayor rapidez y sencillez de programación

respecto a los competidores, por ejemplo Java. Además, su curva de aprendizaje es muy buena. No necesita de frameworks adicionales para su funcionamiento ni de compilación, esto permite desarrollos ágiles. Terminar comentando que para la base de datos se ha elegido MySQL [10], el motivo es que conozco bien el lenguaje y tiene una gran comunidad de soporte, sin comentar por supuesto que es una de la más usadas por muchas organizaciones modernas.

5.2 Seguimiento

Para comprobar el avance continuo del proyecto, y tener un histórico de desarrollo se utilizará Git [11] y Bitbucket [12]. El repositorio público con el código es el siguiente: <https://bitbucket.org/estiif/coolpresentations-agplv3>. Estas herramientas ofrecen un repositorio que registra los cambios en código y ayudan a documentarlos. Además se realizará una sesión semanal con el director del proyecto para estar al tanto de los avances y posibles desviaciones.

Para la monitorización y posibles dudas se ha utilizado correo electrónico y en caso de que fuese necesario, una llamada de audio mediante Hangouts [13]. Además, debido a que se sigue una metodología Kanban usaremos el software Trello [14] que nos facilita una pizarra o “dashboard” en la que podemos organizar y visualizar diversas tareas en paralelo. En la figura 1 tenemos un recorte de la interfaz.

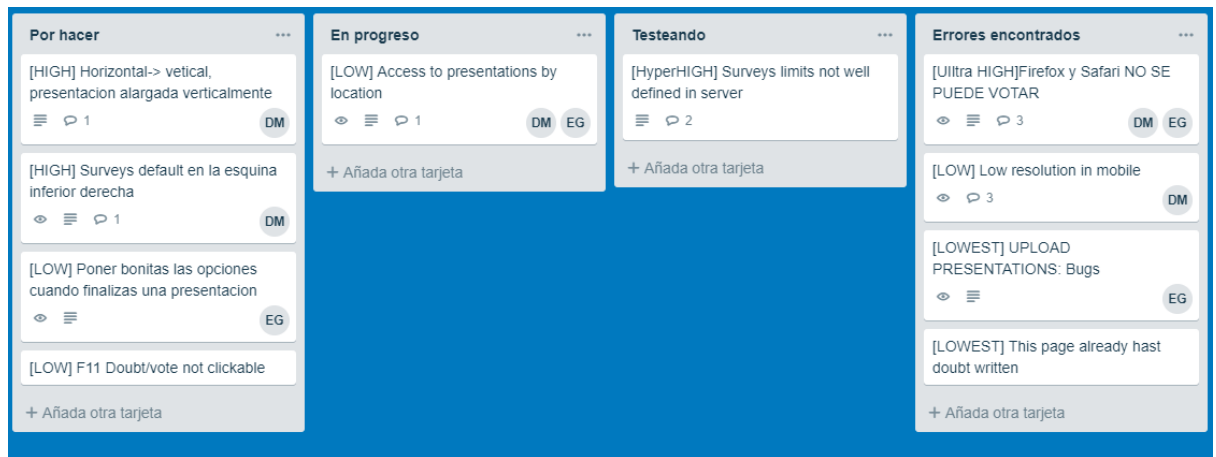


Figura 1: Recorte del tablón de Trello

5.3 Validación de resultados

Gracias al uso de git y con la reunión semanal con el profesor, podemos con facilidad validar gran parte de los avances, sin embargo, también utilizaremos herramientas externas para ello. Se utilizará un RoboTest [15] de código libre que trackea la web y detecta errores. Utilizaremos juegos de pruebas unitarios para comprobar que los nuevos módulos funcionan correctamente.

Además, como sabemos, la mejor forma de validar los resultados es en entornos reales. Para ello se buscarán voluntarios que acostumbren a dar presentaciones/clases para probar la herramienta. Un buen ejemplo de voluntarios serían los mismos profesores de la universidad o compañeros de trabajo que acostumbren a dar presentaciones.

5.4 Obstáculos y riesgos

En las siguientes secciones intentaremos prever posibles riesgos que amenazan el desarrollo del proyecto. Además, se identificará la solución que se planea tomar para cada uno de ellos.

5.4.1 Bugs

El mayor riesgo del proyecto aparece con los bugs o errores imprevistos. Como sabemos tenemos un tiempo limitado para el desarrollo del proyecto, y el tiempo de resolución de bugs no es negligible. Para lidiar con este problema, se preverá un tiempo margen para la resolución de bugs en el planning temporal con cada una de las funcionalidades.

5.4.2 Bajo presupuesto

Aunque a primera vista no parezca un problema grave para una aplicación web lo cierto es que sí. Hoy en día todo servicio online debe tener unos tiempos de respuestas aceptables para satisfacer a los clientes y usuarios. Tener éstos tiempos no se basa simplemente en tener un código eficiente, debemos contar con otros factores.

Los “hosting” o alojamientos web no son gratuitos. El precio depende de varios factores como por ejemplo si la página es estática o dinámica, el número de servicios contratados e incluso el número de visitas. Considerar también que no todas las empresas de “hosting” ofrecen el mismo rendimiento. Normalmente las suscripciones son mensuales o anuales.

Esta herramienta almacena presentaciones para después replicarlas en muchos usuarios. Es por ello que para tener una buena calidad de servicio necesitamos contratar los servicios de una CDN o red de distribución de contenido.

Estás son conjuntos de servidores repartidos geográficamente, que permiten tener un servicio de alta disponibilidad y rendimiento mediante réplicas del contenido.

Para solucionar con estos problemas, se ofrecerá el servicio de manera escalada, es decir, ofreciendo el servicio a una pequeña comunidad para después ir escalando el servicio progresivamente junto a los beneficios. Además se utilizará el servicio DigitalOcean [16] para el “hosting” y a “CloudFlare”[17] como CDN ya que son los que disponen de la mejor relación calidad servicio y precio para pequeños desarrollos.

6 Planificación temporal

En las siguientes secciones se pretende explicar el plan temporal que se llevará a cabo para asegurar la realización del proyecto. Este plan temporal incluirá posibles desviaciones y las afectaciones de éstas. Debemos tener en cuenta que este plan temporal ha sufrido modificaciones a lo largo del proyecto, puesto que se definió en fases iniciales y ha sido alterado, sin embargo, se explicarán las desviaciones tomadas para llegar a la fecha de entrega con el proyecto finalizado.

Para la realización del proyecto, disponemos de aproximadamente 4 meses. Desde 17 de septiembre de 2018 a 1 de febrero de 2019. Este tiempo incluye el estudio de la viabilidad, contextualización, planificación del proyecto e incluso la redacción de este documento. Como se comentará más adelante el tiempo de realización se ha alargado hasta abril de 2019.

6.1 Descripción de las tareas

A continuación describiremos las distintas tareas que se tendrán que llevar a cabo para la realización del proyecto. Además se identificarán posibles dependencias entre ellas y se clasificarán según el nivel de riesgo para la realización del proyecto. Las clasificaremos en tres niveles; bajo, medio o alto. Añadir también que se identificarán los recursos necesarios para cada una de ellas.

6.1.1 Estudio viabilidad

El estudio de la viabilidad y contexto del proyecto ya se ha realizado. En ella se ha estudiado si las características del proyecto cumplen con los requerimientos de la universidad. Ha sido necesario estudiarlo con anterioridad para asegurarnos que el proyecto fuera aceptado por la universidad. Esta primera tarea se ha llevado a cabo

conjuntamente al director del proyecto para asegurar que se ha realizado correctamente, puesto que tiene un nivel alto de riesgo para el proyecto. En cuanto a los recursos necesarios para el desarrollo de esta tarea, únicamente se han necesitado de recursos humanos, el director y yo mismo.

6.1.2 Estudio contexto, planificación temporal y costes

Esta tarea incluye el estudio de la contextualización del proyecto y su alcance. Además pretende desarrollar una planificación temporal que deberá servir de pauta para cumplir con las fechas límites. Otra de los objetivos de esta tarea es calcular los costes que supone realizar el proyecto. Finalmente, esta tarea ayuda a los estudiantes a empezar a escribir un documento, que nos servirá de base, para conseguir una buena memoria del proyecto. Para la realización de esta tarea, a parte de los recursos humanos para escribir el documento, necesitamos de un PC con un editor de texto.

6.1.3 Preparación entorno

La preparación de las herramientas necesarias para el desarrollo del proyecto constituye una tarea en sí. El objetivo del proyecto es desarrollar un servicio web y se puede realizar sobre cualquier S.O, es por ello que lo primero que necesitamos es instalar y configurar Windows 10 puesto que ha sido el SO elegido. Seguidamente necesitaremos instalar el editor de código, como comentamos en nuestro caso Visual Code. Para poder visualizar el desarrollo del servicio web necesitamos de distintos navegadores web, necesitaremos más de uno ya que pueden haber diferencias entre distintos navegadores y nuestro objetivo es que funcione en todos ellos correctamente. En este caso los seleccionados han sido Google Chrome [18], Safari [19] y Firefox [20]. Añadir también que en esta tarea incluimos el tiempo de desarrollar la base de la página web sobre la cual iremos añadiendo funcionalidades.

Es en esta tarea también, dónde debemos contratar un Hosting y configurarlo. Como se ha comentado DigitalOcean ha sido el elegido, y deberemos alquilar los servidores necesarios y configurarlos para nuestras necesidades. Para la realización de esta tarea simplemente necesitaremos recursos humanos para la instalación del Software necesario y el pc de la tarea anterior. Después de instalar todo el software necesario, configurarlo y comprobar que todo funciona correctamente, podemos proceder con la tarea principal de nuestro proyecto, el desarrollo.

6.1.4 Desarrollo del proyecto

Debido al gran peso de esta tarea se subdividirá en otras para ganar granularidad. He decidido subdividirlo por funcionalidades del proyecto, es decir, cada funcionalidad corresponderá a una sub-tarea, de esta manera podremos controlar con mayor facilidad posibles desviaciones y plantear más fácilmente las acciones a tomar frente a estas desviaciones. Además, podremos calcular más precisamente el tiempo que queremos dedicar a cada funcionalidad. Dicho esto, encontramos que de esta tarea aparecen 11 sub-tareas que cada una corresponde con una de las funcionalidades mencionadas. Añadir también que incluiremos una última tarea de optimización con el uso de la CDN. Para la realización de esta tarea necesitaremos recursos humanos para el desarrollo y en cuanto a recursos materiales únicamente el PC de las tareas anteriores.

6.1.5 Verificación y finalización

Una vez acabado todo el desarrollo, deberemos comprobar que los resultados son los deseados. En esta última tarea se verificará que todas las funcionalidades funcionen tal y como se espera además de añadir la últimas optimizaciones si fuesen necesarias. Finalmente, en esta tarea prepararemos el entregable final y la presentación del proyecto.

6.2 Planificación y tiempo estimado

La siguiente tabla muestra el listado de tareas y el tiempo estimado de realización para cada una de ellas.

Tareas	Tiempo estimado (horas)
Estudio viabilidad	15
Estudio contexto, planificación temporal y costes	30
Preparación entorno	60
Acceso a una presentación mediante ubicación	15
Acceso a una presentación mediante contraseña	5
Acceso a una presentación mediante rango horario	15
Acceso mediante combinación	20
Seguimiento presentación dispositivos móviles	40
Presentaciones descargables	15
Encuestas en tiempo real	60
Anotación de dudas	60
Recordatorio de dudas	15
Ranking respuestas	15
Encuesta satisfacción	45
Optimizaciones con CDN	30
Verificación y finalización	60
Total	500

Tabla 1: Tiempo estimado por tarea

6.3 Diagrama de Gantt

Para interpretar el diagrama correctamente, figura 2, debemos tener en cuenta que se trabajarán 30 horas semanales. Si calculamos el tiempo del proyecto con 30 horas semanales obtenemos: 30 horas/semana * 17 semanas = 510 horas. Aclarar también que todas las tareas son realizadas por mí mismo.

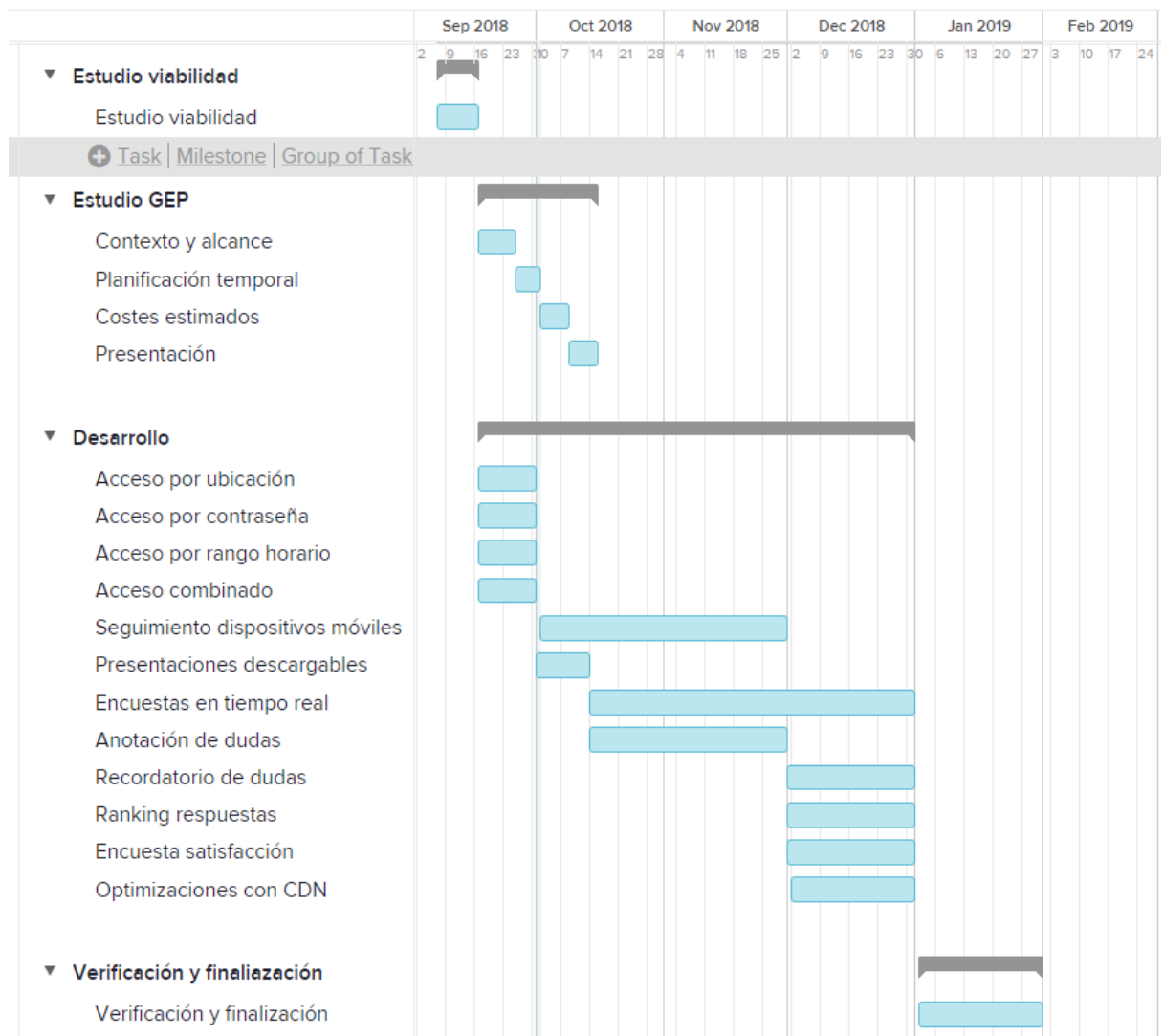


Figura 2: Tiempo estimado por tarea, Gantt

6.4 Desviaciones y plan de acción

En esta sección intentaremos identificar posibles desviaciones a la planificación temporal y el plan de acción que se seguirá en caso de que se dieran. El principal riesgo del proyecto, como comentamos, aparece en los posibles bugs y el tiempo empleado en resolverlos. Para ello se ha estimado un 20% más de margen temporal para cada una de las funcionalidades. Estimamos que el tiempo de margen es suficiente para completar el proyecto exitosamente. En caso contrario, suponiendo que los bugs retrasan de manera incontrolable el desarrollo del proyecto, el plan de acción a tomar será reducir el número de funcionalidades a añadir en el servicio web, de esta manera garantizaríamos que el proyecto pudiera llevarse a cabo aun no teniendo todas las características. Las funcionalidades que se eliminarían dependerán del retraso ocasionado por los bugs y del tiempo estimado de realización para dicha funcionalidad.

Otra posible desviación podría producirse por un mal cálculo del tiempo estimado para cada funcionalidad. En caso de que se produjese, el plan de acción a seguir sería el mismo, es decir, se reducirían el número de funcionalidades para poder seguir asegurando la integridad del proyecto.

6.5 Modificaciones planificación temporal, desviaciones

Esta sección pretende describir las desviaciones y toma de decisiones llevada a cabo a partir de la planificación inicial. Además explicará el impacto en la planificación final.

El principal cambio que ha sufrido la planificación ha sido a nivel temporal. En la planificación temporal se marcaron un número de horas para cada una de las tareas. Los cambios se han producido por dos motivos: El primero, que no llegué a realizar el

número de horas por tarea dentro del plazo marcado, y el segundo, que la estimación de horas era optimista.

El motivo por el cual no se llegó a realizar el número de horas por tarea dentro del plazo marcado ha sido por un cambio de prioridades, ya que encontré una oferta laboral la cual acepté. Para lidiar con este imprevisto sin que afectara al resultado final tuve que posponer el proyecto un cuatrimestre. La nueva planificación temporal ajustada a las nuevos periodos de entrega se pueden ver en el siguiente diagrama de Gantt, figura 3. El número de horas para las tareas sigue siendo el mismo.

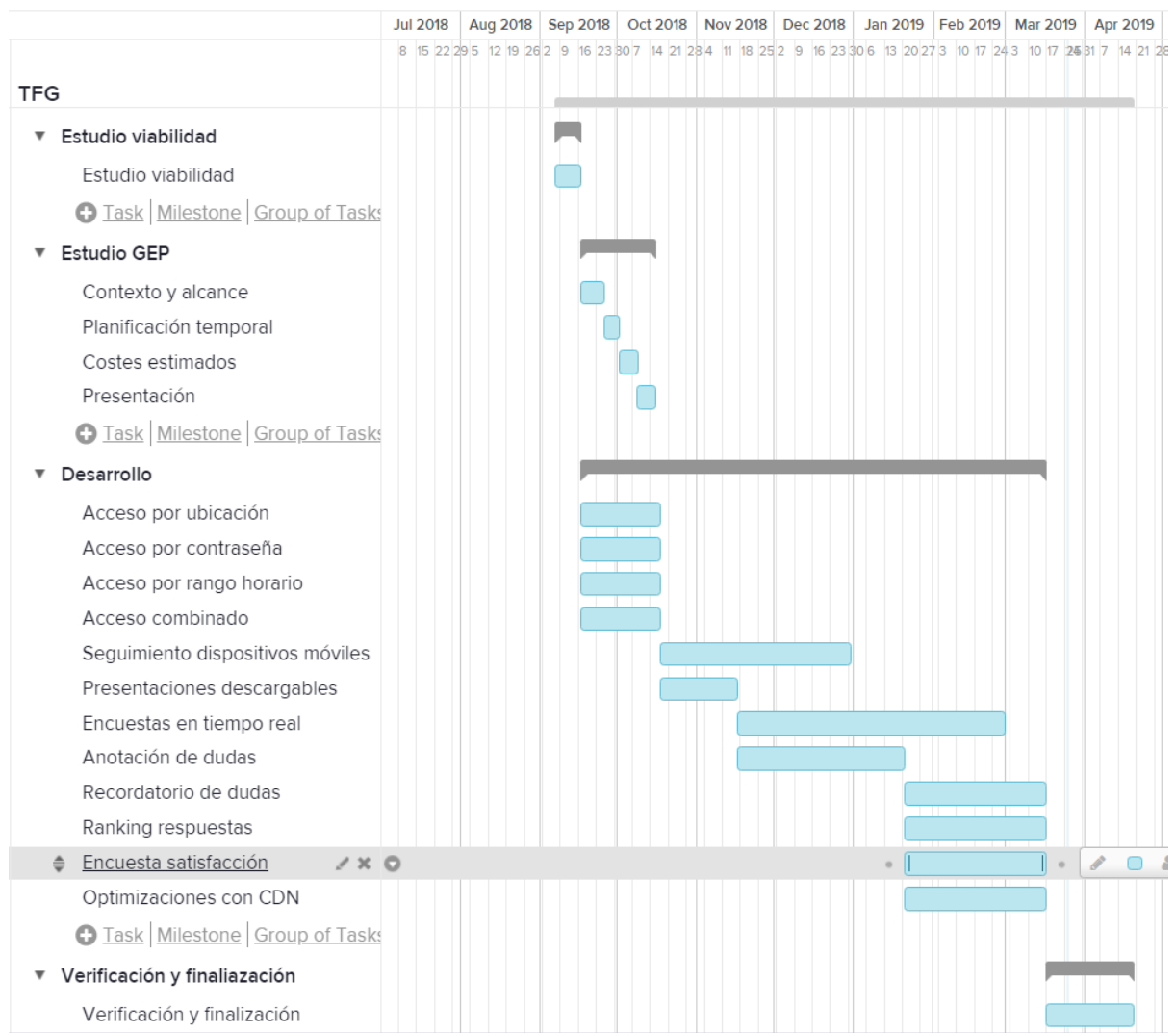


Figura 3: Tiempo estimado por tarea, Gantt

Esta variación de la planificación inicial no afecta a los objetivos del proyecto, ya que las tareas a realizar siguen siendo las mismas. En cuanto a los costes tampoco impacta, ya que el número de horas de trabajo y el gasto de recursos sigue siendo el mismo.

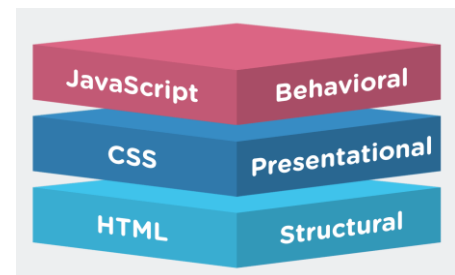
7 Implementación

Las siguientes secciones pretenden explicar cómo se ha llevado a cabo la implementación del proyecto. Se dividirá en capa presentación y capa de datos, además, explicará los algoritmos utilizados por las principales funcionalidades.

7.1 Capa presentación

En esta sección describiremos cómo hemos implementado la interfaz gráfica. En todo momento se ha procurado que sea amigable, intuitiva y práctica para el usuario. Uno de los mayores retos de la interfaz gráfica ha sido que fuera completamente responsive para todo tipo de pantallas con el fin de tener una alta accesibilidad. Como he comentado, en las herramientas de desarrollo, los lenguajes elegidos para la capa de presentación han sido HTML, CSS, Javascript y Ajax. Éste no es en sí un lenguaje de programación sino que es la unión de varias tecnologías.

Con HTML he desarrollado la estructura básica del sitio, con CSS he controlado la presentación, formato y el diseño de la aplicación, y finalmente, con la ayuda de Javascript, controlamos el comportamiento de los elementos.



Para conseguir que la capa de presentación fuera dinámica sin necesidad de recargarse continuamente cada vez que realizamos una petición al servidor hemos utilizado AJAX, acrónimo de **A**synchronous **J**avascript **A**nd **X**ML. Esta tecnología nos ha ayudado a que la aplicación fuera dinámica ya que mantiene una comunicación

asíncrona con el servidor en segundo plano. La figura 4 ilustra el funcionamiento de AJAX:

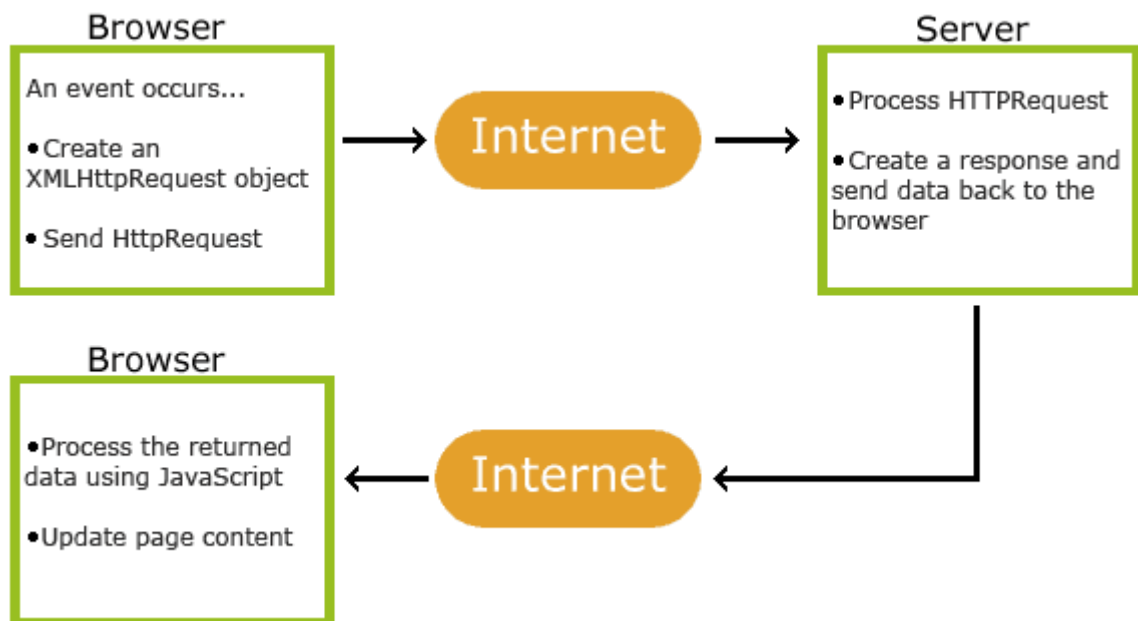


Figura 4: Esquema funcionamiento AJAX

Este tipo de aplicaciones corren en el navegador, es decir, en el lado del cliente, con la ayuda de AJAX conseguimos solicitar nuevo contenido al servidor mediante un evento Javascript. En nuestro caso, las peticiones AJAX llaman a servicios en PHP los cuales le entregan el contenido del servidor. El contenido solicitado se carga en segundo plano sin interferir en la visualización ni en el comportamiento de la aplicación.

A continuación, en las figuras 5,6 ,7 y 8, aparecen algunos pantallazos de la aplicación en la que se observa la interfaz gráfica y su adaptación a distintos tipo de pantallas.

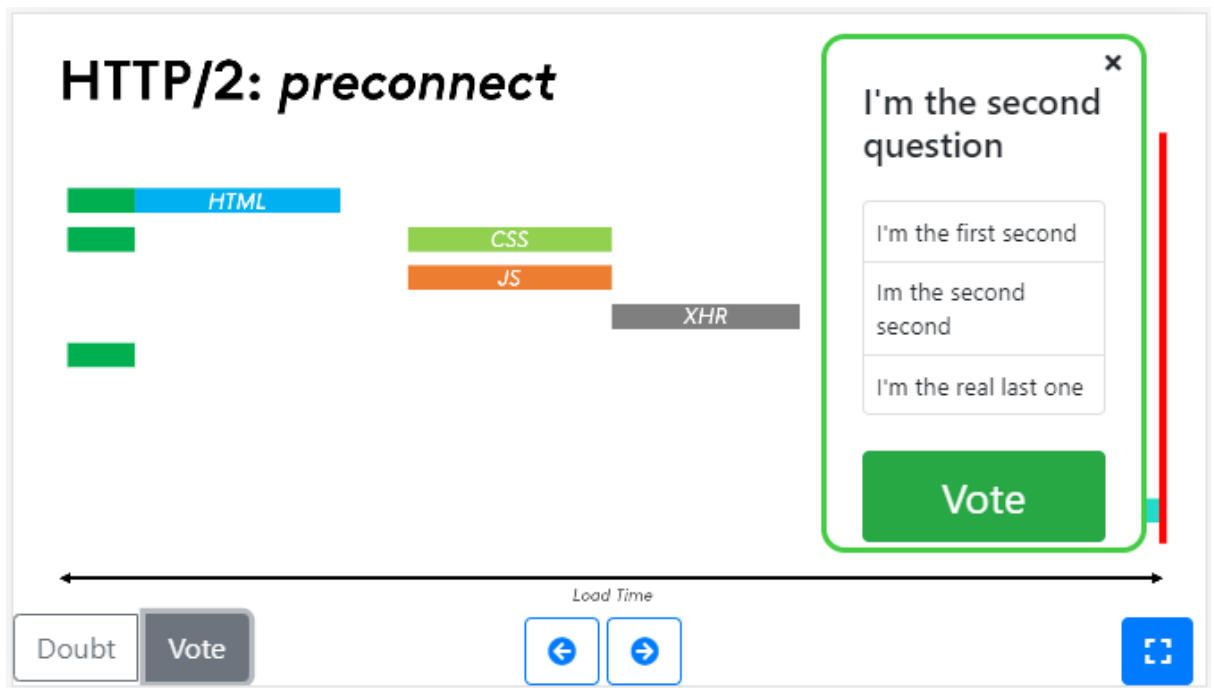


Figura 5: Pantallazo de una presentación en Iphone 6 con orientación horizontal

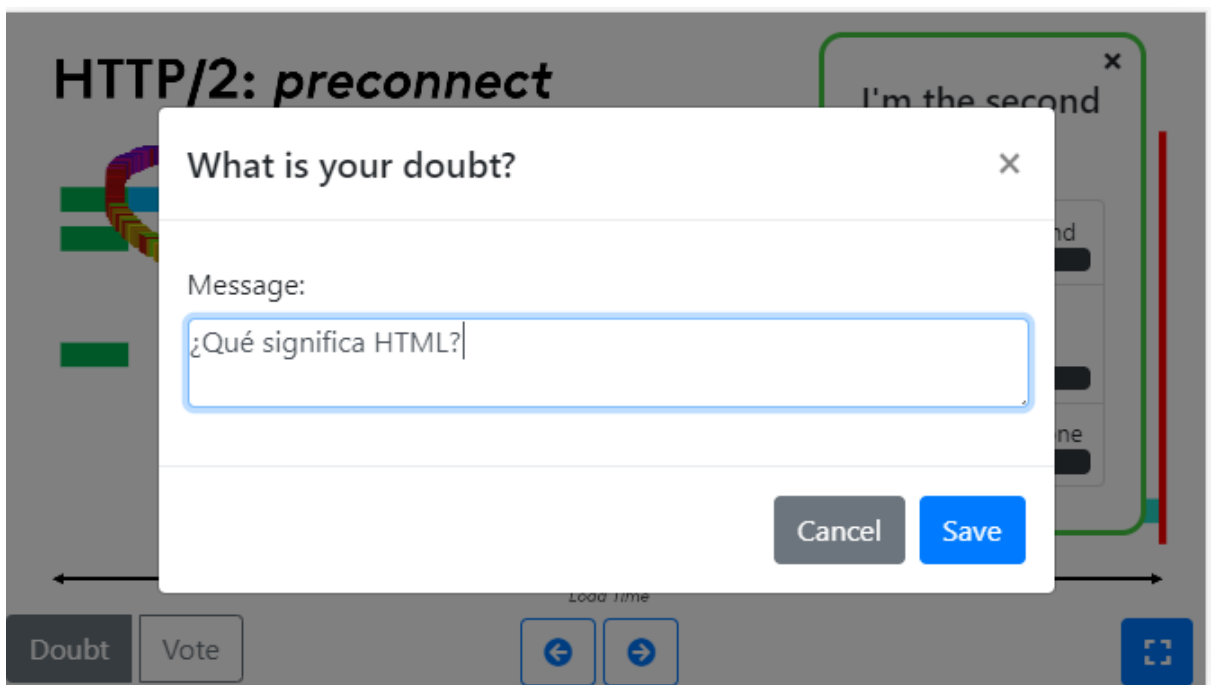


Figura 6: Pantallazo de una presentación en Iphone 6 con orientación horizontal, modal de dudas.

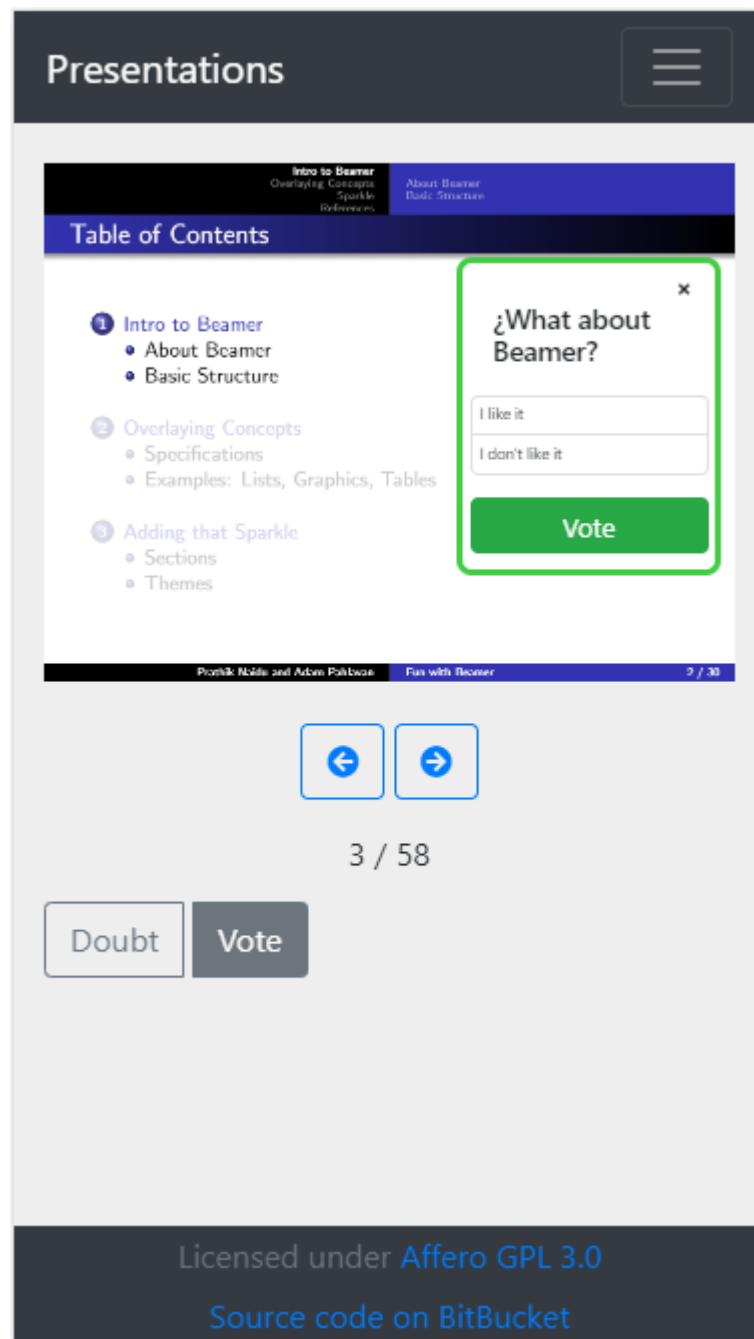


Figura 7: Pantallazo de una presentación en Iphone 6 con orientación vertical, sin pantalla completa

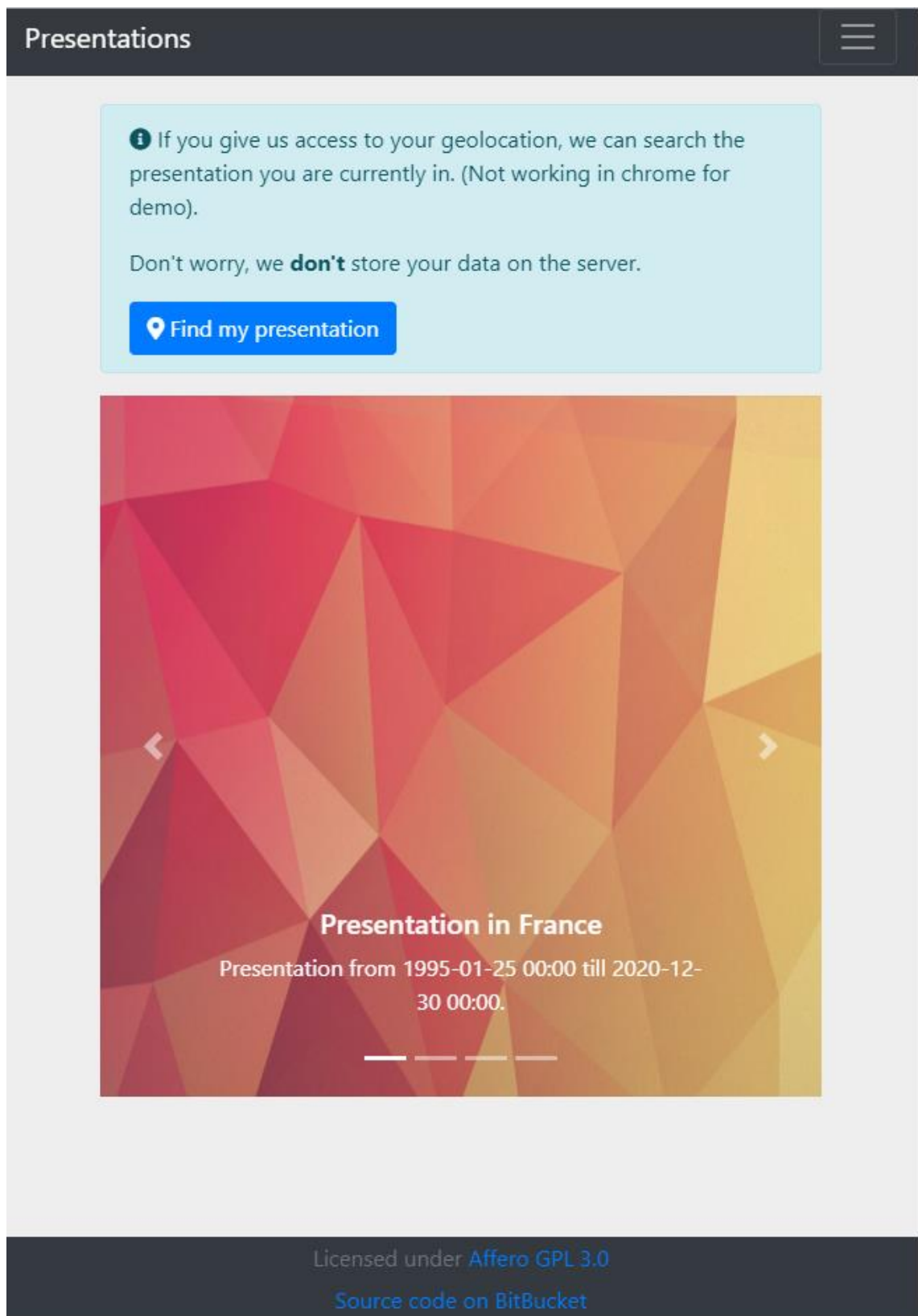


Figura 8: Pantallazo de la portada de la aplicación en pc

7.1.1 Librerías utilizadas

Para facilitar el desarrollo de la capa de presentación haremos uso de algunas librerías css y javascript. Éstas son estáticos cacheables que nos ahorrarán gran parte de implementación. Entre las librerías encontramos:

- **Bootstrap**: librería de código abierto que ayuda a diseñar sitios web responsive. Contiene plantillas de tipografía, formularios, botones... Es el segundo proyecto más destacado en GitHub y es usado por la NASA [21]. Basada en CSS y JS.
- **FontAwesome**: librería muy útil para añadir iconos en webs, basada en CSS y LESS [22]. Se incluye en BootstrapCDN.
- **CookieConsent**: librería de código abierto que nos ayuda a cumplir la RGPD añadiendo en nuestro portal el aviso de cookies. Basada en CSS y JS.
- **jQuery**: librería de código abierto. Permite simplificar la manera de interactuar con los documentos HTML, manipular el DOM, manejar eventos, desarrollar animaciones y agregar interacción con AJAX. Basada en JS.
- **interact.js**: librería de código abierto. Facilita la implementación de interacciones como arrastrar y soltar, gestos multitáctiles... Se adapta bien a todo tipo de pantallas pero presenta algunas incompatibilidades con Safari y Firefox. Basada en JS.
- **PDF.js**: librería de código abierto. Facilita la implementación de un visor de PDF[23] en HTML. Liderada por Mozilla [24]. Basada en JS.

- **Leaflet**: librería de código abierto. Se utiliza para crear aplicaciones de mapas web. Basada en Javascript. Básicamente se ha utilizado en el proyecto para seleccionar la localización dónde se dará la presentación.



Figura 9: Logo de las 7 librerías respectivamente

7.2 Capa de datos

En la siguiente sección describiremos cómo se ha implementado la capa de datos y cómo se accede a los datos almacenados, en este caso, en una base de datos MySQL mediante PHP.

Gracias a una herramienta escrita en PHP podremos manejar la administración de la base de datos mediante un navegador web. Esta herramienta es phpMyAdmin [25], figura 10. Con ella podemos visualizar los datos insertados, ver los distintos esquemas, ejecutar sentencias SQL, exportar la base de datos... y todo ello desde el propio

navegador. Dado que la capa de datos está implementada en PHP, no ha sido necesario instalar PHP expresamente en el servidor para usar esta herramienta.

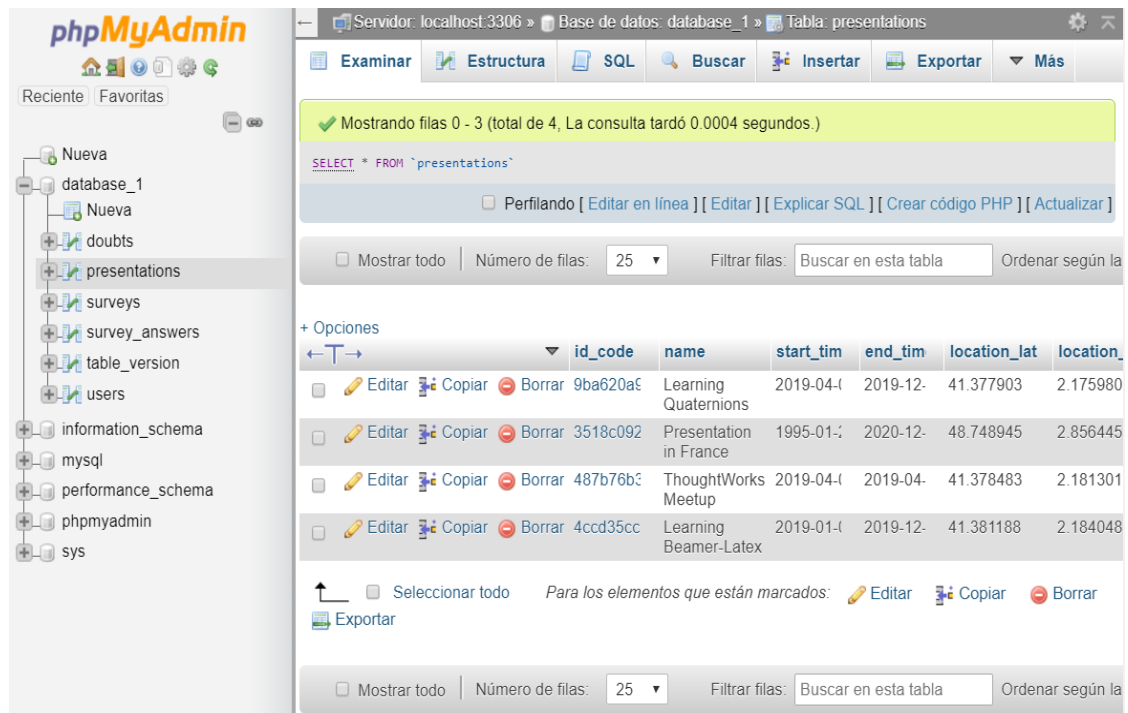


Figura 10: Vista navegador del frontal de phpMyAdmin

Siempre que accedemos a la base de datos intentamos que las conexiones sean seguras. Además, intentamos controlar posibles errores en los datos de entrada y en los devueltos por la base de datos. Para ejemplificar mejor, a continuación, explicaremos la función de inserción de usuario y contraseña cuando un nuevo usuario se registra en la aplicación. En la figura 11 podemos ver el código de la función

```

36 function save_into_database($mysqli) {
37     $nick = $_POST['nick'];
38     $pass = $_POST['pass'];
39     $randomPseudoBytes = bin2hex(openssl_random_pseudo_bytes(32, $strong));
40     if( !$strong ) {
41         http_response_code(500);
42         $mysqli->close();
43         die('openssl_random_pseudo_bytes not strong enough.');
```

Figura 11: Recorte de la función “save_into_database”

- **líneas 37-38:** recogemos los datos introducidos por el usuario, los cuales, nos llegan mediante AJAX.
- **línea 39:** generamos una cadena de bytes pseudo-aleatorio.
- **líneas 40-44:** el siguiente IF comprueba que se usó un algoritmo criptográficamente fuerte para producir los bytes pseudo-aleatorios.
- **línea 45:** utilizamos la función “**hash**”, la cual genera un valor cifrado con base a un string. Como vemos, el string es una unión entre el resultado de ejecutar “**uniqid**” y el la cadena de bytes pseudo-aleatoria. “**uniqid**” genera un ID único prefijado basado en la hora actual en microsegundos. El resultado de ejecutar “**hash**” lo recogemos en la variable **\$salt**.
- **línea 46:** Volvemos aplicar “**hash**”, pero esta vez el string utilizado es la unión de la variable **\$salt** anterior y el password que introdujo el usuario. Con esto conseguimos eliminar la posibilidad de que el resultado pueda buscarse a partir

de una lista de pares precalculados de hash y sus entradas originales, conocidas como tablas rainbow [26].

- **línea 47:** preparamos la sentencia SQL que ejecutaremos en la base de datos, pero sin pasarle los parámetros.
- **línea 48:** pasamos los parámetros a la sentencia SQL mediante la función “**bind_param**”. Gracias a esto evitamos inyecciones SQL.
- **líneas 49-61:** Ejecutamos la sentencia SQL y hacemos los pertinentes controles de posibles errores.

Esta patrón de implementación se sigue en todas las funciones que acceden a la base de datos, tanto inserciones como consultas, todos los parámetros de todas las sentencias SQL se pasan mediante la función “**bind_param**”. Los datos más sensibles siempre se guardan aplicando un “**hash**” para no almacenarlos en texto plano.

7.2.1 Estructura base de datos

Tal y como he comentado, he elegido una bases de datos relacional, en este caso, MySQL. Este tipo de bases de datos se componen de varias tablas, denominadas relaciones. No pueden existir dos tablas con el mismo nombre ni registro. Cada tabla es a su vez un conjunto de campos (columnas) y registros (filas).

He estructurado la base de datos en 6 tablas:

- **table_version:** Su finalidad es saber si estamos trabajando con la última versión de la base de datos. Simplemente cuenta con un entero que indica el número de versión de la base de datos.

- **presentations:** Se encarga de almacenar la información básica de las presentaciones, como el nombre de la presentación, la contraseña de acceso, el lugar dónde se dará, la fecha de inicio y final...
- **surveys:** Almacena la información relacionada con las encuestas que aparecen sobre las presentaciones, por ejemplo, la pregunta que se plantea, la página en la que debe aparecer, el tamaño y posición de la encuesta...
- **survey_answers:** Contiene las respuestas de las encuestas, podemos encontrar las posibles respuestas, los votos recibidos por respuesta, la página y presentación dónde deben aparecer...
- **doubts:** Almacena las dudas escritas por los usuarios, podemos encontrar el usuario que escribió la duda, si es que estaba registrado, la sesión del navegador y la página y presentación dónde se anotó.
- **users:** Información de usuario, únicamente guardamos el nombre de usuario, el id que le asignamos, su contraseña encriptada y el "salt" con el que encriptamos.

A continuación, en la figura 12 podemos ver un esquema de todas las tablas con todos sus campos y su tipo. Los campos que tienen el icono de una pequeña llave amarilla significa que son claves primarias, es decir, ese campo no podrá tener registros iguales ya que sirven para identificar de manera única un registro. Las flechas indican claves foráneas, es decir, identifica una columna o grupo de columnas en una tabla que se refiere a una columna o grupo de columnas en otra tabla.

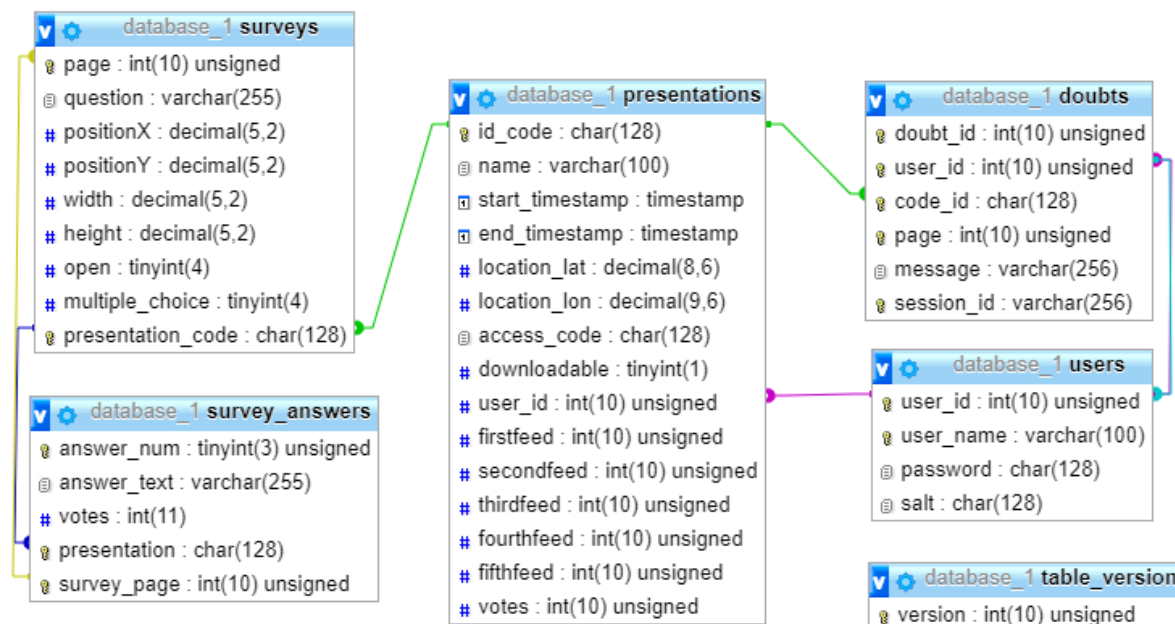


Figura 12: Esquema de las tablas y campos de la base de datos

Cada vez que tenía que hacer un cambio no planificado en la base de datos, generaba un script de actualización y generaba un nuevo número de versión. A continuación, en la figura 13, podemos ver un ejemplo:

```
-- Select the database
USE database_1;

-- Insert version
UPDATE `table_version` SET `version`=4 WHERE `version`=3;

-- Add width and height to surveys
ALTER TABLE `surveys` ADD `width` DECIMAL(5,2) NOT NULL DEFAULT '50.00';
ALTER TABLE `surveys` ADD `height` DECIMAL(5,2) NOT NULL DEFAULT '50.00';
```

Figura 13: Sentencias para actualizar la base de datos de la versión 3 a la 4

También hay preparado un script de generación de la base de datos, el cual está actualizado a la última versión, a continuación, en la figura 14, un ejemplo de la creación de la tabla 'presentations':

```

-- Presentations table
-- It stores the information relative to the presentations in the system
CREATE TABLE IF NOT EXISTS `presentations` (
  `id_code` CHAR(128) NOT NULL,
  `name` VARCHAR(100) NOT NULL,
  `start_timestamp` TIMESTAMP NOT NULL,
  `end_timestamp` TIMESTAMP, -- NULLABLE
  `location_lat` DECIMAL(8,6), -- NULLABLE
  `location_lon` DECIMAL(9,6), -- NULLABLE
  `access_code` CHAR(128) DEFAULT NULL, -- NULLABLE
  `downloadable` TINYINT(1) DEFAULT 0 NOT NULL,
  `user_id` INT UNSIGNED NOT NULL,
  `firstfeed` INT UNSIGNED NOT NULL DEFAULT 0,
  `secondfeed` INT UNSIGNED NOT NULL DEFAULT 0,
  `thirdfeed` INT UNSIGNED NOT NULL DEFAULT 0,
  `fourthfeed` INT UNSIGNED NOT NULL DEFAULT 0,
  `fifthfeed` INT UNSIGNED NOT NULL DEFAULT 0,
  `votes` INT UNSIGNED NOT NULL DEFAULT 0,

  PRIMARY KEY (`id_code`),

  FOREIGN KEY (`user_id`)
    REFERENCES `users` (`user_id`)
    ON UPDATE CASCADE
    ON DELETE RESTRICT
);

```

Figura 14: Sentencias para crear la tabla 'presentations'

7.3 Funciones / Algoritmos principales

El objetivo de esta sección es describir algunos algoritmos y funciones que creo que son relevantes para el desarrollo del proyecto. Los primeros hacen referencia a la capa de presentación y los últimos a la capa de datos. No documentamos todos los algoritmos y funciones ya que sería muy extenso y porque hay muchas similitudes entre algunos de ellos.

7.3.1 Añadir dudas

La posibilidad de poder añadir dudas sobre las diapositivas es uno de los objetivos principales del proyecto. Para conseguir esta característica haremos uso de la librería `interact.js`, de los modales de `bootstrap` y de `AJAX`. Como comento en el manual de usuario que aparece en el apéndice, si dibujamos con el cursor, o con el dedo en los dispositivos táctiles, un círculo sobre la parte de la diapositiva dónde tenemos

dudas, se nos mostrará un modal que nos permite anotar dudas. En este apartado únicamente describiré como se ha implementado la aparición del modal cuando el círculo se dibuja.

La librería `interact.js` nos proporciona mecanismos para simplificar la implementación de esta funcionalidad. En nuestro caso los pasos que realizamos son los siguientes:

- 1) Indicar a qué elemento del HTML aplicaremos las interacciones. En nuestro caso es un canvas que tiene la clase `“.rainbow-pixels-canvas”`, línea 7 figura 15.
- 2) Mientras se esté realizando el arrastrable. Líneas 14-39 figura 15.
 - i) Calculamos el color del cuadrado/pixel a pintar dependiendo de la velocidad y el ángulo de movimiento. Líneas 17 a 24 figura 15.
 - ii) Nos guardamos la posición en la que empieza el arrastrable. Líneas 26-31 figura 15.
 - iii) Dibujamos cuadrados que simulan un trazo. Líneas 33 a 39, figura 15.
- 3) Cuando termina el arrastrable hacemos el cálculo de si se ha acabado en el mismo punto en el que se empezó, con un cierto margen de error marcado por la variable `“variability”`. En caso afirmativo simulamos un click encima del botón que abre el modal. El botón está oculto para el usuario en el HTML.

```

5 | var pixelSize = 10;
6 | var variability=52;
7 | interact(".rainbow-pixel-canvas")
8 |   .origin("self")
9 |   .draggable({
10 |     max: Infinity,
11 |     maxPerElement: Infinity,
12 |   })
13 |   // draw colored squares on move
14 |   .on("dragmove", function(event) {
15 |     var context = event.target.getContext("2d"),
16 |         // calculate the angle of the drag direction
17 |         dragAngle = 180 * Math.atan2(event.dx, event.dy) / Math.PI;
18 |     // set color based on drag angle and speed
19 |     context.fillStyle =
20 |       "hsl(" +
21 |         dragAngle +
22 |         ", 86%, " +
23 |         (30 + Math.min(event.speed / 1000, 1) * 50) +
24 |         "%)";
25 |     //Calculate first and last
26 |     if (firstcircle==0){
27 |       if (firstsquareX==0){
28 |         firstsquareX=event.pageX - pixelSize / 2;
29 |         firstsquareY=event.pageY - pixelSize / 2;
30 |       }
31 |     }
32 |     // draw squares
33 |     context.fillRect(
34 |       event.pageX - pixelSize / 2,
35 |       event.pageY - pixelSize / 2,
36 |       pixelSize,
37 |       pixelSize
38 |     );
39 |   })
40 |   // Open Doubts Modal
41 |   .on("dragend", function(event) {
42 |     if ((firstsquareX >= ((event.pageX - pixelSize / 2)-variability)) &&
43 |         (firstsquareX <= ((event.pageX - pixelSize / 2)+variability))){
44 |       if ((firstsquareY >= ((event.pageY - pixelSize / 2)-variability)) &&
45 |           (firstsquareY <= ((event.pageY - pixelSize / 2)+variability))){
46 |         if (firstcircle==0){
47 |           $("#dudabutton").trigger('click');
48 |           firstcircle=1;
49 |         }
50 |       }
51 |     }
52 |   })

```

Figura 15: Recorte de código que genera la apertura del modal de dudas

7.3.2 “Near Real Time” resultado encuestas, AJAX vs WebSockets

La posibilidad de responder encuestas y ver los resultados en tiempo real es uno de los objetivos principales del proyecto. Para conseguir que los resultados se mostrarán en tiempo real, o, casi tiempo real, también conocido como “near real time NRT” teníamos dos opciones. La primera de ellas es utilizar WebSockets [27]. Éstos proporcionan un canal bidireccional y full-duplex sobre un único socket TCP [28], además están diseñados para ser implementados en navegadores y servidores web. Están orientados a tener contenido interactivo en tiempo real, con latencias muy bajas, durante largos periodos, es decir mantienen una conexión directa con el usuario, bidireccional, durante un periodo de tiempo, por ejemplo para un videojuego online o para una aplicación en que varias personas dibujan a la vez. Cómo vemos no es exactamente nuestro caso ya que sólo interactuamos para enviar datos al servidor una vez, cuando votamos, pero sí que necesitamos ir consultando si se han actualizado los resultados. Además, no necesitamos latencias bajas ya que no influye en el funcionamiento de la aplicación.

La segunda opción era utilizar AJAX para conseguir “Near Real Time” usando la técnica de “polling” [29]. Esta técnica se basa en realizar peticiones asíncronas al servidor usando AJAX cada cierto tiempo. Esto encaja más en nuestra aplicación, ya que no necesitamos una conexión abierta durante largos periodos de tiempo, sino que, únicamente la necesitamos mientras se esté mostrando una diapositiva que contenga una encuesta. Además, tampoco requerimos bajas latencias, no hay problemas en que la votación de otro usuario tarde 5 segundos en poder visualizarse. Otro de los motivos por el cual me decanté por esta técnica es que no tenía que incorporar más tecnologías a la aplicación, AJAX ya lo estaba usando durante todo el proyecto. A continuación, figura 16, podemos observar la función de AJAX que realiza la técnica de “polling”:

```
479 // Updates the survey information for the displayed survey
480 function surveyUpdateCallback() {
481     if( !currentSurvey ) {
482         return;
483     }
484     var theSurvey = currentSurvey;
485     $.get('survey_info.php',
486         {'pres_id': PRES_CODE,
487          'page': currentSurvey.page},
488         (data) => {
489             for(var key in data) {
490                 theSurvey[key] = data[key];
491             }
492             surveyAnswersDraw(theSurvey);
493             setTimeout(surveyUpdateCallback, 5000);
494         },
495         'json');
496 }
```

Figura 16: Recorte de la función AJAX que implementa “polling”

Como podemos observar, es una función recursiva que se llama a sí misma. El primer IF, de la línea 481 comprueba que estamos en una diapositiva que contiene una encuesta, es decir, no saldremos de esta función hasta que pasemos de diapositiva. En la línea 493 podemos ver la llamada recursiva, la cual espera 5 segundos antes de volver a llamarse, esto implica que los resultados de las encuestas se verán actualizados cada 5 segundos.

7.3.3 Uso de plantillas html y javascript

Con el objetivo de generar contenido html dinámicamente dado un evento javascript haremos uso de plantillas html las cuales iremos insertando y borrando mediante javascript. Esta técnica aparece en varias partes de la aplicació, a continuación describiremos una que es extrapolable a todas las demás. Como sabemos, la estructura de los resultados de las encuestas dependen de su contenido,

por ejemplo de si la encuesta tiene 3 o 4 posibles respuestas. A continuación, figura 17, una imagen para recordarlo:

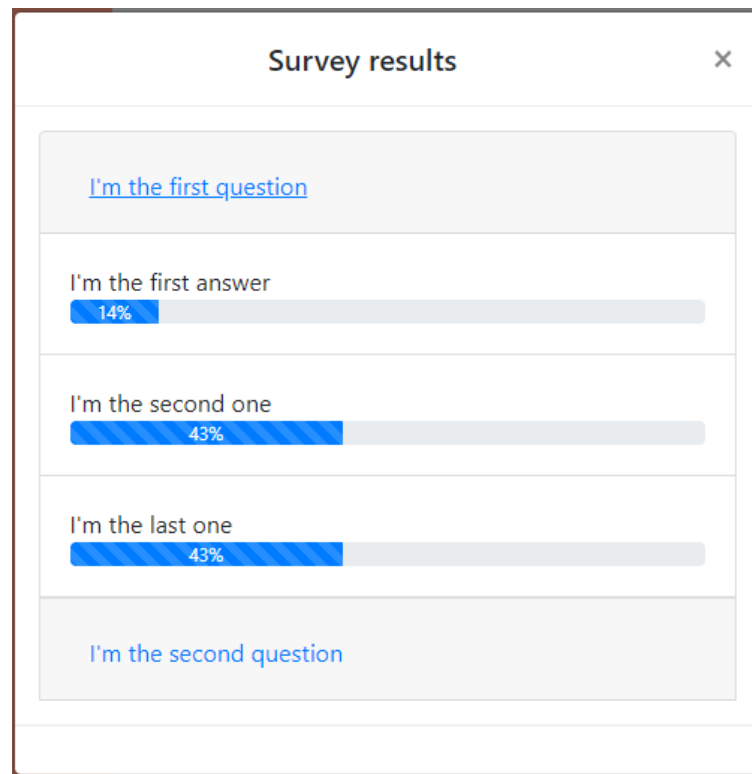


Figura 17: Recorte del modal de resultados de encuestas

Como podemos ver, en azul aparecen las cuestiones planteadas por la encuesta. En este caso, la presentación tenía sólo dos. Si apretamos en la cuestión se despliegan las respuestas y el porcentaje de voto que ha recibido cada una. Aquí podemos diferenciar dos plantillas html distintas. La primera plantilla genera los desplegables de las cuestiones y la segunda plantilla genera las respuestas dentro de los desplegables. A continuación, en la figura 18, podemos observar las dos plantillas respectivamente:

```

<script type="text/html" id="results-questions-template">
  <div class="card">
    <div class="card-header" id="headingOne">
      <h2 class="mb-0">
        <button class="btn btn-link" type="button" data-toggle="collapse"
          data-target="###IDENTIFICADORES%" aria-expanded="true" aria-controls="collapseOne">
          %%QUESTION%%
        </button>
      </h2>
    </div>
  </div>
  <div id="###IDENTIFICADORES%" class="collapse" aria-labelledby="headingOne" data-parent="#accordionExample">
  </div>
</script>
<script type="text/html" id="results-answers-template">
  <div class="card-body">
    <span>%%ANSWERS%%</span>
    <div class="progress">
      <div class="progress-bar progress-bar-striped progress-bar-animated" role="progressbar"
        aria-valuenow="%%PERCENTAGE%%" aria-valuemin="0" aria-valuemax="100">
        %%PERCENTAGE%%
      </div>
    </div>
  </div>
</script>

```

Figura 18: Recorte de código en el que aparecen plantillas HTML

En el código podemos ver las dos plantillas separadas por los tags `<script>` y `</script>`. Además, también observamos que hay varias cadenas de caracteres delimitadas por `%%--%%`. Estos delimitadores nos ayudan a cambiar el valor de la cadena de caracteres vía Javascript, así podremos controlar el contenido que mostramos. En el siguiente recorte podemos ver cómo reemplazamos estas cadenas de caracteres por los valores que necesitamos para generar el contenido:


```

var ResultsQuestionTemplate = $('#results-questions-template').text();
var ResultsAnswersTemplate = $('#results-answers-template').text();
for(let k=0; k<pagesurveys.length; k+=1){
  let htmlString = "";
  htmlString = ResultsQuestionTemplate
    .replace("%%QUESTION%%", SURVEYS[pagesurveys[k]].question)
    .replace("%%IDENTIFICADORES%%", identificadores[k])
    .replace("%%IDENTIFICADORES%%", identificadores[k]);
  $('#.accordion').append(htmlString);
  let sum = SURVEYS[pagesurveys[k]].answers.reduce((prev,next)=>prev+parseInt(next.votes), 0);
  let resanswers = [];
  for(let i=0; i<SURVEYS[pagesurveys[k]].answers.length; i+=1){
    let percentage = Math.round(parseInt(SURVEYS[pagesurveys[k]].answers[i].votes) / sum * 100);
    let htmlStringA = $(ResultsAnswersTemplate
      .replace("%%PERCENTAGE%%", percentage)
      .replace("%%PERCENTAGE%%", percentage)
      .replace("%%ANSWERS%%", SURVEYS[pagesurveys[k]].answers[i].text));
    htmlStringA.find(".progress-bar").css("width", percentage+"%");
    resanswers.push(htmlStringA);
  }
  $('#.collapse').last().append(resanswers);
}

```

Figura 19: Recorte de código en el que aparece la inserción de plantillas HTML

En las dos primeras líneas recogemos en dos variables Javascript las plantillas HTML de la figura 18. El primer bucle itera una vez por cada diapositiva que contenga una encuesta. Podemos ver que mediante la instrucción “.replace” cambiamos el contenido de las cadenas delimitadas con %%--%% por el valor que deseamos. Una vez hecho el “replace” con la instrucción de la línea 9, es decir con “append” insertamos la variable que contiene la plantilla ya con los valores cambiados en el lugar que toque del DOM. El siguiente bucle, dentro del anterior, itera una vez por cada respuesta que contenga la encuesta de la iteración. Como podemos apreciar aplicamos la misma técnica, es decir, con la instrucción “.replace” cambiamos el valor de las plantillas y con “.append” añadimos el contenido al DOM.

7.3.4 Comprobación de usuario y contraseña

El acceso al contenido de los usuarios es delicado y hay que securizarlo para evitar posibles ataques. Este acceso se realiza logueandose en la aplicación mediante el nombre de usuario y la contraseña correspondiente. Por este motivo, describiré

como se ha realizado la comprobación de usuario y contraseña y finalmente cómo se otorga el acceso a la aplicación con los datos de usuario.

La validación la realizamos en dos partes, primeramente en el lado del cliente con Javascript y finalmente en el lado del servidor mediante PHP. En el lado de cliente realizamos dos acciones, empezamos comprobando que los datos introducidos cumplen con una expresión regular, para encontrar texto de acuerdo a un patrón, y finalmente hasheadamos la contraseña antes de enviarla a la base de datos. En la figura 20 y 21 podemos verlo.

```
function onClickPre(url) {  
    errorsAlert.hide();  
    let nick = form.find("#nick").val().trim();  
    let pass = form.find("#pwd").val().trim();  
    let remember = (form.find("#remember").prop("checked")?true:false);  
    if(checkValues(nick, pass)) {  
        // hash pass even before sending (sha.js)  
        var shaObj = new jsSHA("SHA-512", "TEXT");  
        shaObj.update(pass);  
        pass = shaObj.getHash("HEX");  
        connection(url, nick, pass);  
    }  
}
```

Figura 20: Recorte de código JS que realiza la conexión al servidor

Podemos ver que antes de realizar la conexión mediante AJAX ejecutamos la función “checkValues”. En caso positivo hasheadamos la contraseña y la enviamos al servidor hasheada y no en texto plano.

```

function checkValues(nick, pass) {
  let html = "";
  if(nick === "" || pass === "") {
    html = "Please, introduce name and password.";
    errorsAlert.html(html).show().fadeOut(5000);
    return false;
  }

  let nickValid = nick.match(/^[A-Za-z0-9_\-\.]+$/);
  let passValid = (pass.length >= 8);

  if(!nickValid)
    html += "<strong>Nickname:</strong> the nickname can only contain alphanumeric characters, '_'";
  if(!passValid)
    html += "<strong>Password:</strong> the password should be, at least, 8 characters long.";

  if(!nickValid || !passValid)
    errorsAlert.html(html).show();

  return nickValid && passValid;
}

```

Figura 21: Recorte de código JS que comprueba los datos introducidos por el usuario

La función “checkValues” de la figura 21 comprueba que el nombre de usuario introducido no sea nulos y que cumpla con la expresión regular de la línea 9. Además comprueba que la contraseña tenga un mínimo de 8 caracteres.

Una vez tenemos los datos en el servidor lo primero que hacemos es comprobar que los datos que nos llegan cumplen con las condiciones que hemos implementado en JavaScript. Para ello, lo primero que hacemos es ejecutar la función “check_name_and_pass” en PHP . En la figura 22 podemos ver la función.

```
13 function check_name_and_pass($mysqli) {  
14     if(!isset($_POST['nick'], $_POST['pass'])) {  
15         http_response_code(400);  
16         $mysqli->close();  
17         die('Error: nick and pass not received.');
```

Figura 22: Recorte de código PHP de la función “check_name_and_pass”

Como podemos ver la función consta de tres condicionales if. El primero comprueba que llegue el usuario y contraseña al servidor. El segunda comprueba que el usuario cumpla con la misma expresión regular que utilizamos en la figura 21 en Javascript. Finalmente, el tercer condicional comprueba que la contraseña que llega al servidor este hasheada, esto lo conseguimos aplicando la expresión regular de la línea 26.

Una vez hemos verificado que los datos que llegan al servidor son correctos hacemos una consulta a la base de datos para recuperar la contraseña y el “salt” con el que se hasheo en el registro. Ahora ya podemos aplicar un hash usando el “salt” recuperado de la base de datos y con la contraseña que hemos recibido por el usuario (recordemos que ya una vez hasheada) . Si este último resultado del hash concuerda con la contraseña recuperada de la base de datos podemos afirmar que la contraseña de registro y con la que intenta loguearse son iguales y procedemos a iniciar la sesión del usuario. En la figura 23 podemos ver el código que describo en este párrafo.

```

function login($mysqli) {
    $nick = $_POST['nick'];
    $pass = $_POST['pass'];

    $stmt = $mysqli->prepare('SELECT user_id, password, salt FROM users WHERE user_name=?');
    $stmt->bind_param('s', $nick);
    if(!$stmt->execute()) {
        http_response_code(500);
        $stmt->close();
        $mysqli->close();
        die('Error in the query '.$stmt->errno);
    }
    $stmt->bind_result($id,$realPass,$salt);
    if(!$stmt->fetch()) {
        http_response_code(400);
        $stmt->close();
        $mysqli->close();
        die('Unknown user or password');
    }
    $stmt->close();

    $pass = hash('sha512', $pass.$salt); // We mix it with the salt :)
    if($pass !== $realPass) {
        http_response_code(400);
        $mysqli->close();
        die('Unknown user or password');
    }

    // So we finally logged in
    $_SESSION['user_id'] = $id;
    $_SESSION['user'] = $nick;
    //$_SESSION['session_token'] = $token; no session tokens yet
}

```

Figura 23: Recorte de código PHP de la función “login”

7.3.5 Obtener presentaciones a partir de una condición

Una de las funciones que más se utiliza en el proyecto es la de consultar en la base de datos qué presentaciones tenemos disponibles a partir de una condición dada. Estas condiciones pueden variar mucho, ya sea obtener presentaciones por ubicación, por rango horario, por usuario... Encontramos que habían muchas variantes y hacer una función por cada condicional distinto implicaba un código muy extenso y poco práctico. Es por este motivo que se ha implementado una función de consulta a la cual se le pasa como parámetros los condicionales, es decir el “where” de la sentencia SQL, los parámetros de la sentencia y el tipo de los parámetros. Con esta información es suficiente para generar la sentencia SQL. En la figura 24 podemos observar la función:

```

function get_presentations_where($where, $paramTypes, $params) {
    global $mysqli;

    $presentations = [];

    $stmt = $mysqli->prepare(
        'SELECT id_code,name,start_timestamp,end_timestamp,location_lat,
        |location_lon,downloadable,user_id,          You, a few seconds ago • Uncommitted changes
        IF(access_code IS NULL, 0, 1) AS access_code_required
        FROM presentations WHERE '.$where);
    if(!$stmt) {
        throw new Exception('Error in the question query preparation ' . $mysqli->error, $mysqli->errno);
    }
    try {
        call_user_func_array([$stmt, 'bind_param'],array_merge([$paramTypes],$params));
        if (!$stmt->execute()) {
            throw new Exception('Error in the question query ' . $stmt->error, $stmt->errno);
        }
        $stmt->bind_result($presentationCode,$name,$start,$end,$lat,$lon,$download,$userId,$req_code);
        while($stmt->fetch()) {
            $presentations[] = [
                'id_code' => $presentationCode,
                'name' => $name,
                'start_timestamp' => $start,
                'end_timestamp' => $end,
                'access_code' => $req_code?true:false,
                'location' => ['lat'=>$lat, 'lon'=>$lon],
                'downloadable' => $download?true:false,
                'author' => $userId
            ];
        }
    } finally {
        $stmt->close();
    }
    return $presentations;
}

```

Figura 24: Recorte de código en el que aparece la función “get_presentations_where”

Podemos observar cómo se prepara la sentencia SQL a ejecutar y cómo recibe por parámetros las condiciones “where” que le llegan a la función. También podemos ver, en la primera línea del bloque try, como se llama a la función “bind_params” con los parámetros y tipos que llegan como parámetros a la función, mediante la función “call_user_func_array”. Finalmente recogemos los resultados devueltos por la base de datos, cerramos la conexión y devolvemos las presentaciones obtenidas.

7.3.6 Index.php protección contra ataques

Cuando un usuario accede a coolpresentations.me Apache recibe la petición y la redirige hacia el index.php. Al ser este fichero el punto de entrada, se ha implementado una capa de seguridad para controlar cuáles son las únicas páginas que se puedan cargar en nuestro cuerpo HTML, evitando así posibles ataques.

En la figura 25 podemos observar el código que implementa esta capa de seguridad. En la línea 8 declaramos un array de pares, clave valor, que indica cuales son las únicas páginas permitidas para cargarse. En las líneas 10-13 declaramos variables que nos servirán para configurar el directorio de scripts, páginas html y estilos css. En la línea 15 consultamos la página actual, la cual la obtenemos de la misma URL. En las siguientes líneas comprobamos que la variable recibida por la url es una de las claves del array de pares. En caso afirmativo, cargamos los recursos necesarios.


```
1 <?php
2 // Php session start
3 session_start();
4
5 // $pages maps the page key to the page files names, this are the only
6 // allowed pages to load into the body (preventing attacks)
7 // Please notice the key 'invalid' is reserved for invalid keys introduced by the client
8 $pages = ['ini'=>'inicio', 'view'=>'viewer', 'log'=>'login', 'upload'=>'upload_pdf', 'mypres'];
9 // Some variables to configure the pages directory, styles and scripts file
10 $pages_dir = '../pages/';
11 $pages_scripts_file_sufix = "_scripts";
12 $pages_styles_file_sufix = "_styles";
13 $pages_php_scripts_file_sufix = "_php_scripts";
14 // We get the current page from the GET variables, or set a default one
15 $current_page_key = (isset($_GET['p'])?$_GET['p']:'ini');
16 // Look for the page and save $current_page_file if it is right
17 $current_page_file = "";
18 if(array_key_exists($current_page_key, $pages)) {
19     $current_page_file = $pages[$current_page_key];
20 } else {
21     $current_page_key = 'invalid'; // invalid key
22 }
23
24 if($current_page_file != "") {
25     $scripts_file = $pages_dir.$current_page_file.$pages_php_scripts_file_sufix.'.php';
26     if(file_exists($scripts_file)) {
27         include $scripts_file;
28     }
29 }
30 ?>
```

Figura 25: Recorte de código PHP en el que se implementa la capa de seguridad

Para ejemplificar mejor el funcionamiento de este código, en la figura 26 podemos ver la url del navegador para acceder a la página de login y en la figura 27 un ejemplo de una url que no tiene una clave de página correcta.

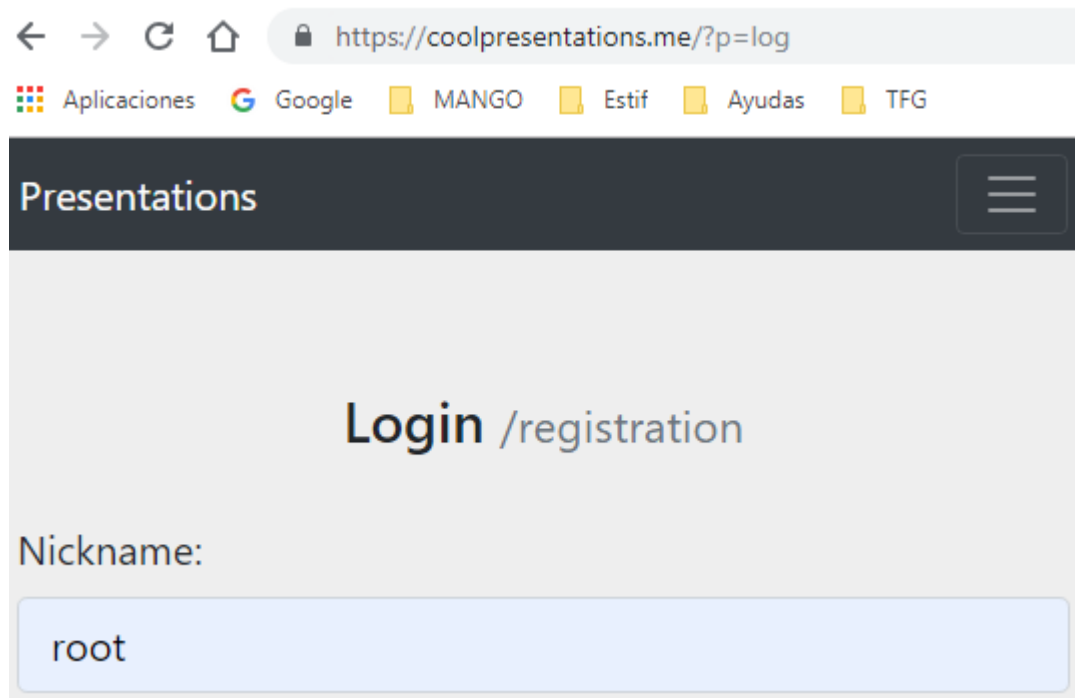


Figura 26: Recorte del navegador en el que se observa una url con clave correcta

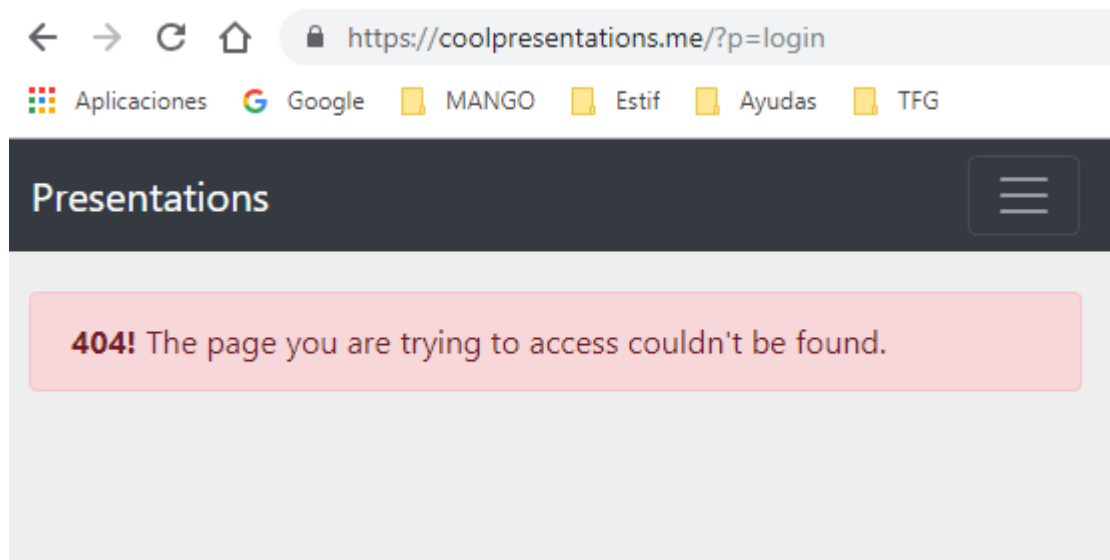


Figura 27: Recorte del navegador en el que se observa una url con clave incorrecta


8 Servidor y calidad de servicio

En las siguientes secciones explicaremos cómo se ha configurado el servidor para alojar la aplicación web, además, describiremos los servicios utilizados para conseguir una buena disponibilidad y accesibilidad. En la configuración del servidor incluimos la monitorización y seguridad del mismo, así como la instalación del entorno necesario y la gestión de espacio en disco.

8.1 Droplet, DigitalOcean

Cómo sabemos, comprar un servidor no es una acción trivial, ya que supone un coste muy elevado, es por ello, que diversas empresas ofrecen alquiler de servidores, también conocidos como servidores en cloud. El principal beneficio es que únicamente pagas por consumo, no tienes que pagar el hardware, además es fácilmente escalable y replicable sin mencionar que presentan una alta accesibilidad y confiabilidad. Las mejores empresas, bajo mi punto de vista, son AWS (Amazon Web Services)[30], GPC (Google Cloud Platform) [31], Heroku [32] y DigitalOcean.

Después de comparar distintos proveedores cloud me decanté por DigitalOcean. El motivo es que tengo créditos gratuitos gracias al “Student Developer Pack” de GitHub, el cual, te regala créditos en distintas empresas si eres universitario. Una vez solicitados los créditos gratuitos levanté un “Droplet” de DigitalOcean. “Droplet” es el nombre con que DigitalOcean llama a sus servidores virtuales, es decir, su servicio de computación. El “Droplet” lo situé en Amsterdam ya que es un punto bastante céntrico en Europa y cerca de España, con el fin de conseguir una buena accesibilidad. A continuación, podemos observar un recorte de la interfaz de DigitalOcean dónde se observa el “Droplet” con sus características.




CoolPresentations DEFAULT
Class project / Educational purposes

[Resources](#) [Activity](#) [Settings](#)

DROPLETS (1)

• 💧 **CoolPresentations** master

Image	 Ubuntu18.04 x64	Region	AMS3
Size	1 vCPUs 2GB / 50GB Disk (\$10/mo) Resize	IPv4	188.166.110.116
		IPv6	Enable
		Private IP	Enable

DOMAINS (1)

coolpresentations.me	1 A / 3 NS / 1 SOA
--------------------------------------	--------------------

Figura 28: Recorte de la interfaz de DigitalOcean en el que aparece las características del “Droplet”

Como vemos en la figura 28, la imagen seleccionada para el servidor ha sido Ubuntu 18.04 [33]. La elección se debe a que estoy más familiarizado con este sistema operativo y además es de código abierto. También podemos ver que tiene una única CPU, 2GB de memoria RAM y 50GB de almacenamiento en un disco SSD, todo ello por el módico precio de 10 dólares al mes, sin contar que nos reservan una IP pública fija para el “Droplet”, en este caso la 188.166.110.116. Finalmente podemos ver que tenemos un dominio configurado apuntando hacia el servidor, hablaremos de esto más adelante.

Para acceder al servidor lo hacemos mediante el comando `ssh` (Secure Shell) [34]. Cuando creamos el “Droplet” la propia interfaz de DigitalOcean nos permite añadir un certificado digital el cual podemos usar posteriormente para conectarnos de manera segura a nuestro “Droplet”. Para la transferencia de archivos utilizamos el comando `scp` (Secure Copy) [35]. En este comando también podemos hacer uso del certificado para validar el acceso. No encontré la necesidad de implementar un sistema de integración continua, por ejemplo usando Jenkins [36] junto con Bitbucket, ya que trabajaba yo sólo y en local y cuando conseguía algún progreso lo subía simplemente haciendo uso de “`scp`”. Una vez copiados los ficheros en el servidor purgaba el contenido cacheado por la CDN y ya podía observar los cambios.

Con el objetivo de tener un stack LAMP (Linux, Apache, MySQL, PHP) [37] he instalado Apache, PHP y MySQL en el servidor. Para que no se llene el espacio de disco debido a logs hacemos uso de “`logrotate`” [38] con el cual podemos rotar los logs cómo queramos, ya sea eliminándolos o enviándolos algún tipo de almacenamiento.

8.1.1 Apache

El punto de entrada hacia nuestra aplicación es un servidor web Apache. La elección se debe a que es de código libre, compatible con plataformas Unix [39] y que dispone de una gran comunidad para dar soporte. Otro de los motivos es que me decidí por un stack LAMP ya que es muy común y podría encontrar mucha información al respecto en caso de encontrar problemas. En la figura 29 observamos la configuración de Apache.

```
ServerTokens Prod
ServerSignature Off
<VirtualHost *:80>
    DocumentRoot /var/www/html/public_html/
    Redirect / https://www.coolpresentations.me
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
<VirtualHost *:443>
    DocumentRoot /var/www/html/public_html/

    <Directory /var/www/html/>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <IfModule mod_dir.c>
        DirectoryIndex index.php
    </IfModule>
</VirtualHost>
```

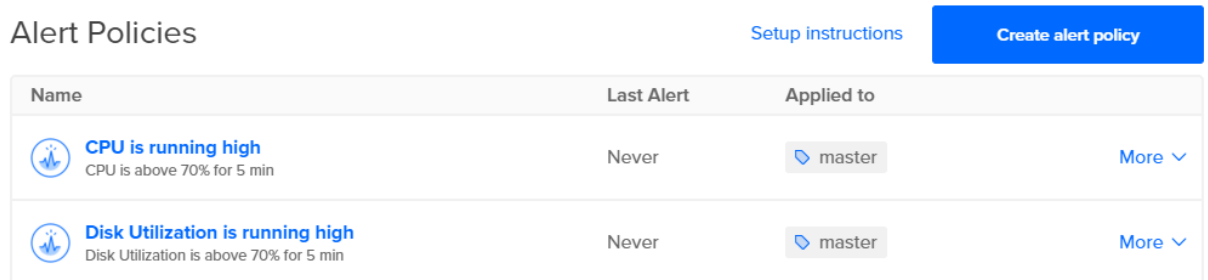
Figura 29: Recorte de la configuración de Apache

Las dos primeras líneas evitan que aparezca la versión del servidor, en los navegadores, cuando sucede un error, por ejemplo, un error 403 de acceso denegado. Podemos ver que todas las peticiones que sean http, es decir puerto 80, son redirigidas hacia https, es decir, puerto 443. Esto lo conseguimos con la instrucción “Redirect” de la línea 5. Evitamos que listen nuestros directorios mediante el navegador ya que no incluimos la instrucción “Indexes”[40] en ninguna opción. Además, marcamos cómo carpeta raíz de apache el directorio principal del proyecto en las líneas 4 y 11.

8.1.2 Monitorización

Para la monitorización del servidor haremos uso de dos herramientas distintas. La primera de ellas es la propia interfaz de DigitalOcean. En ella podemos encontrar gráficas de ancho de banda usado, uso de CPU, lecturas y escrituras en disco por

segundo... Otra opción interesante es la de poder generar alertas en base a unas reglas personalizadas. En nuestro caso he configurado dos, una de CPU que alerta cuando su uso está por encima del 70% durante 5 minutos y otra si el uso de disco también está por encima del 70% durante 5 minutos. Ambas notifican vía mail. En la figura 30 podemos observar la interfaz de DigitalOcean de alertas.







Name	Last Alert	Applied to	
 CPU is running high CPU is above 70% for 5 min	Never	 master	More ▾
 Disk Utilization is running high Disk Utilization is above 70% for 5 min	Never	 master	More ▾

Figura 30: Recorte de la interfaz de Alertas de DigitalOcean

La otra herramienta que usaremos para la monitorización es Prometheus [41], Alertmanager [42] y node-exporter [43]. Node_exporter es una aplicación instalada en el servidor que sirve para publicar métricas del mismo en un endpoint cualquiera por el puerto que deseemos. Con la ayuda de Prometheus, el cual está instalado en mi ordenador personal de casa, podemos recolectar todas estas métricas y generar alertas y reportes. Finalmente, haciendo uso de Alertmanager podemos gestionar que alertas queremos que sean notificadas y por qué medio. En las siguientes figuras, 31, 32 y 33, podemos ver el fichero de configuración de Prometheus para que encuentre el servidor, el fichero de configuración de las alertas y las métricas que publica node_exporter sobre el servidor. Podemos observar que las métricas se publican por el puerto 9100 en el path /metrics. También podemos ver que se han configurado 3 alertas, una para CPU, otra para memoria disponible y finalmente otra para saber si node_exporter está funcionando correctamente.

```

# my global config
global:
  scrape_interval:     60s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 60s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            - alertmanager:9093
# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  - "/etc/prometheus/droplet-alerts.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: 'Droplet'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    tls_config:
      insecure_skip_verify: true
    static_configs:
      - targets: ['188.166.110.116:9100']

```

Figura 31: Fichero configuración prometheus.yml

```

groups:
- name: Droplet-CPU
  rules:
  - alert: Droplet-CPU
    expr: 100-(avg by (instance) (irate(node_cpu{mode="idle",job="Droplet"}[5m])) * 100) > 80
    for: 5m
    labels:
      severity: 1
- name: Droplet-FreeMem
  rules:
  - alert: Droplet-FreeMem
    expr: (100 * (1 - ((avg_over_time(node_memory_MemFree{job="Droplet"}[5m]) + avg_over_time(node_memory_Cached{job="Droplet"}[5m]) + avg_over_time(node_memory_Buffers{job="Droplet"}[5m])) / avg_over_time(node_memory_MemTotal{job="Droplet"}[5m]))) > 80
    for: 5m
    labels:
      severity: 1
- name: Droplet-NodeExporter-UP
  rules:
  - alert: Droplet-NodeExporter-UP
    expr: up{job="Droplet"} < 1
    for: 1m
    labels:
      severity: 1

```

Figura 32: Fichero configuración droplet-alerts.yml


```

go_gc_duration_seconds{quantile="1"} 0.008864799000000001
go_gc_duration_seconds_sum 993.580035184
go_gc_duration_seconds_count 110430
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 26
# HELP http_request_duration_microseconds The HTTP request latencies in microseconds.
# TYPE http_request_duration_microseconds summary
http_request_duration_microseconds{handler="prometheus",quantile="0.5"} NaN
http_request_duration_microseconds{handler="prometheus",quantile="0.9"} NaN
http_request_duration_microseconds{handler="prometheus",quantile="0.99"} NaN
http_request_duration_microseconds_sum{handler="prometheus"} 4.008663689032689e+10
http_request_duration_microseconds_count{handler="prometheus"} 95191
# HELP http_request_size_bytes The HTTP request sizes in bytes.
# TYPE http_request_size_bytes summary
http_request_size_bytes{handler="prometheus",quantile="0.5"} NaN
http_request_size_bytes{handler="prometheus",quantile="0.9"} NaN
http_request_size_bytes{handler="prometheus",quantile="0.99"} NaN
http_request_size_bytes_sum{handler="prometheus"} 2.1235758e+07
http_request_size_bytes_count{handler="prometheus"} 95191
# HELP http_requests_total Total number of HTTP requests made.
# TYPE http_requests_total counter
http_requests_total{code="200",handler="prometheus",method="get"} 95191
# HELP http_response_size_bytes The HTTP response sizes in bytes.
# TYPE http_response_size_bytes summary
http_response_size_bytes{handler="prometheus",quantile="0.5"} NaN
http_response_size_bytes{handler="prometheus",quantile="0.9"} NaN
http_response_size_bytes{handler="prometheus",quantile="0.99"} NaN
http_response_size_bytes_sum{handler="prometheus"} 2.3459031623e+10
http_response_size_bytes_count{handler="prometheus"} 95191
# HELP node_boot_time Node boot time, in unixtime.
# TYPE node_boot_time gauge
node_boot_time 1.545896595e+09
# HELP node_context_switches Total number of context switches.
# TYPE node_context_switches counter
node_context_switches 1.5930348457e+10
# HELP node_cpu Seconds the cpus spent in each mode.
# TYPE node_cpu counter
node_cpu{cpu="cpu0",mode="guest"} 0
node_cpu{cpu="cpu0",mode="idle"} 8.52603972e+06
node_cpu{cpu="cpu0",mode="iowait"} 28625.11
node_cpu{cpu="cpu0",mode="irq"} 0
node_cpu{cpu="cpu0",mode="nice"} 124.48
node_cpu{cpu="cpu0",mode="softirq"} 2182.69
node_cpu{cpu="cpu0",mode="steal"} 4722.62
node_cpu{cpu="cpu0",mode="system"} 105314.73
node_cpu{cpu="cpu0",mode="user"} 687690.87

```

Figura 33: Recorte de algunas métricas publicadas por node_exporter

8.2 CDN, CloudFlare

Con el objetivo de conseguir una buena calidad de servicio haremos uso de una CDN (Content Delivery Network). Éstas son redes de servidores proxy y sus centros de

datos distribuidas geográficamente. El objetivo es proporcionar una alta disponibilidad y un alto rendimiento distribuyendo el contenido para que esté más cerca de los usuarios finales.

La CDN elegida ha sido CloudFlare. El motivo de la elección es que dispone de un plan totalmente gratuito, el cual es más que suficiente. Nos ofrece datos de analítica como número de visitante únicos, número de peticiones, localización de la procedencia de las peticiones... También nos ofrece un servicio de DNS, sin contar toda la seguridad que añade a la aplicación, por ejemplo, evitando ataques DDOS, pues recordemos que las peticiones en primer punto pasarán por los servidores de CloudFlare antes de llegar al origen. Finalmente, ofrece el servicio de cachear contenido, de esta manera conseguimos que los estáticos de la aplicación, como los css o javascripts, estén más cerca del usuario final, y nos ahorramos peticiones al origen, por lo que el servidor necesita menos recursos y por consiguiente menos costes. En las dos siguientes figuras, 34 y 35, podemos ver la interfaz de CloudFlare.

Notifications

Billing update

We are upgrading our billing system. Review answers to common questions.

[Dismiss](#)

Quick Actions

[Purge Cache](#)

[DNS Settings](#)

Under Attack Mode

Show visitors a JavaScript challenge when visiting your site.



Development Mode

Temporarily bypass our cache. See changes to your origin server in realtime.



Como podemos observar, disponemos del modo “Under Attack” el cual si lo activamos muestra una validación insertada por javascript la cual comprueba que el visitante sea fiable. Además, disponemos del modo “development” con el cual el contenido estático no se cachea y podemos observar los cambios en el servidor al momento, en caso de que este modo no estuviera activo, deberíamos purgar el contenido estático cacheado para poder observar los cambios en el servidor.

Figura 34: Recorte de la interfaz de CloudFlare

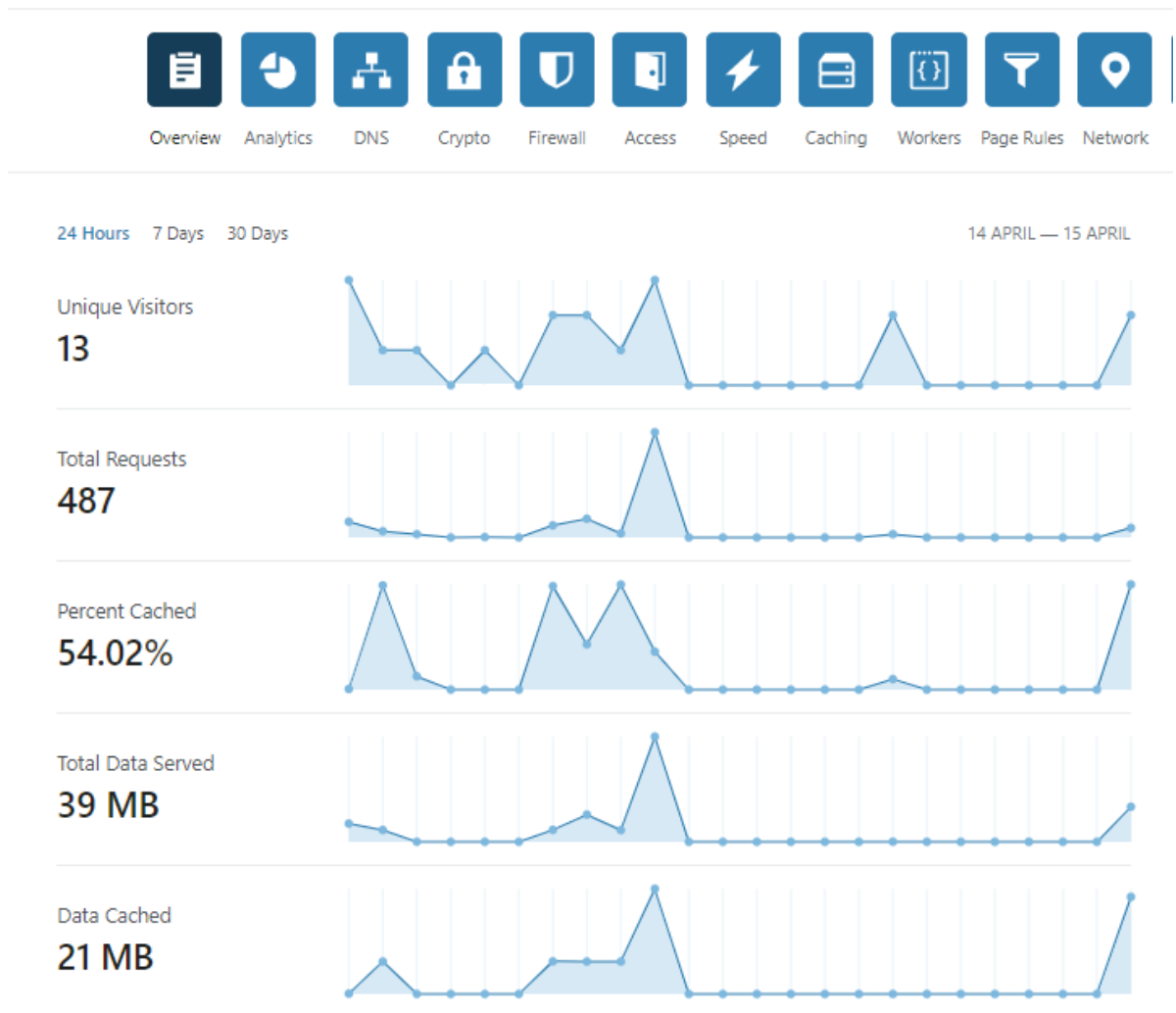


Figura 35: Recorte de la interfaz de CloudFlare

En el recorte anterior, podemos ver 5 gráficas distintas, todas ellas aplican a las últimas 24 horas del momento en que se tomó el recorte. La primera de ellas indica el número de visitantes únicos. La siguiente indica el número de peticiones realizadas. Seguidamente podemos ver el porcentaje del contenido cacheado respecto al servido. La cuarta gráfica nos indica la cantidad de datos servidor y para finalizar la última la cantidad de datos servidos cacheados.

8.3 CA, Let's Encrypt - CertBot

Para que la transmisión de datos entre el servidor y el usuario sea segura haremos uso de un certificado SSL. Esto evita ataques del tipo “man in the middle”. Para conseguir un certificado necesitamos los servicios de una CA (Autoridad certificadora) [44], en este caso la elegida ha sido Let's Encrypt [45]. El motivo de la elección es que es gratuita, de código libre y con mecanismos de automatización para desplegar el certificado y regenerarlos de nuevo.

CertBot [46] es una herramienta que ayuda a automatizar despliegues de certificados de Let's Encrypt. En su página, figura 36, únicamente debemos rellenar dos campos, uno para indicarle el servidor web que utilizamos, en nuestro caso Apache, y otro para indicarle el sistema operativo, es decir, Ubuntu 18.04.

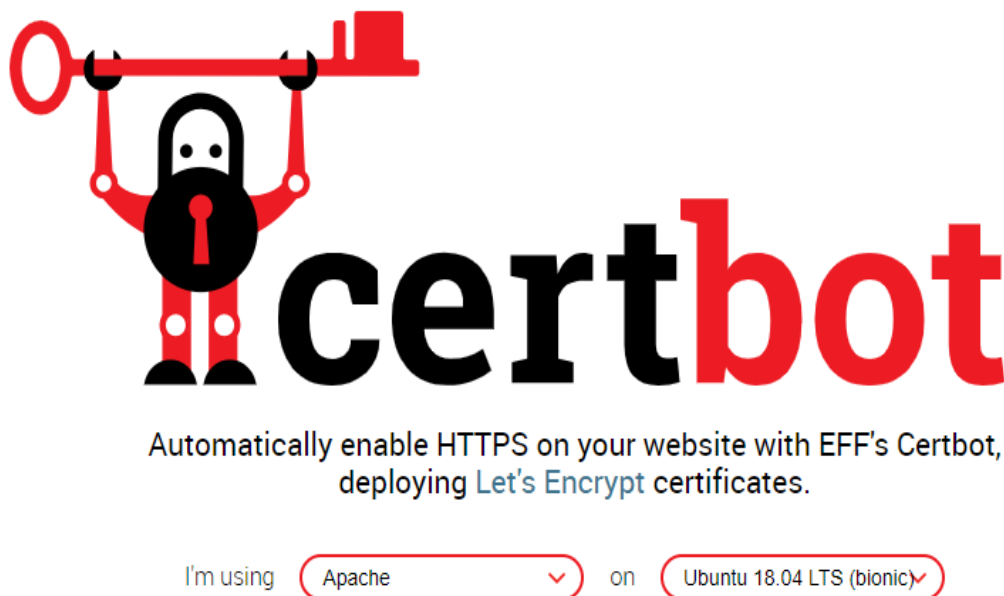


Figura 36: Recorte de la interfaz de Certbot

Una vez rellenados los dos campos se genera una pequeña guía con los comandos necesarios a ejecutar para configurar el certificado. Una vez ejecutados, se ha añadido un certificado al servidor y se ha configurado Apache para que sepa dónde consultarlo. Gracias a esta herramienta es muy fácil generar sitios seguros vía https y regenerar los

certificados automáticamente. A continuación, figura 37, podemos observar el certificado que ve el usuario al acceder al portal:

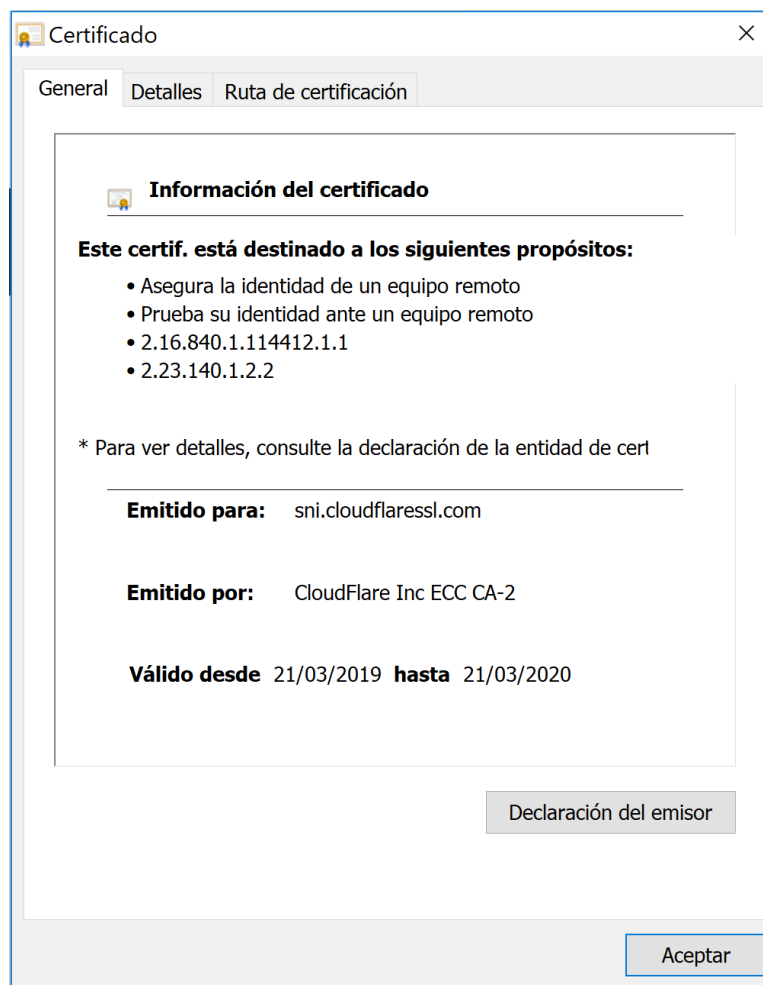


Figura 37: Recorte del certificado entregado por coolpresentations.me

Pero... vemos que el certificado es emitido por CloudFlare, nuestra CDN! y no por Let's Encrypt, la CA, como he estado comentando. ¿Por qué sucede esto? CloudFlare también actúa como CA y él mismo genera los certificados para los dominios que se configuran en su plan gratuito. Los usuarios reciben el certificado de CloudFlare y éste hace uso del certificado de Let'sEncrypt para acceder al servidor. Si el certificado de Let'sEncrypt no funcionase, CloudFlare devolvería un error 500 ya que no sabría si la conexión a origen es segura. En resumen, para el primer tramo de la petición, es decir, del usuario a los servidores de la CDN hacemos uso del certificado de CloudFlare y para

el segundo tramo, es decir de la CDN a origen, hacemos uso del certificado de Let'sEncrypts. Si deseamos utilizar un certificado distinto sin perder el servicio de CloudFlare debemos mejorar nuestro plan a uno de pago. Por este motivo, el certificado que finalmente servimos al usuario es el de CloudFlare.

8.4 Dominio, NameCheap

Coolpresentations.me ha sido el dominio elegido bajo el que nuestra aplicación responde, es el punto de entrada para los usuarios finales. Para ello he adquirido, durante un año, un dominio .me en la empresa registradora de nombres NameCheap [47]. El motivo de la elección de esta empresa registradora es que en el “Student developer pack” de GitHub nos regalan un dominio .me durante un año. Una vez registrado el dominio, hemos delegado su resolución de nombre al DNS de CloudFlare, es allí donde finalmente se traduce a la IP de nuestro servidor. A continuación, figura 38 y 39, observamos dos recortes de la configuración comentada.

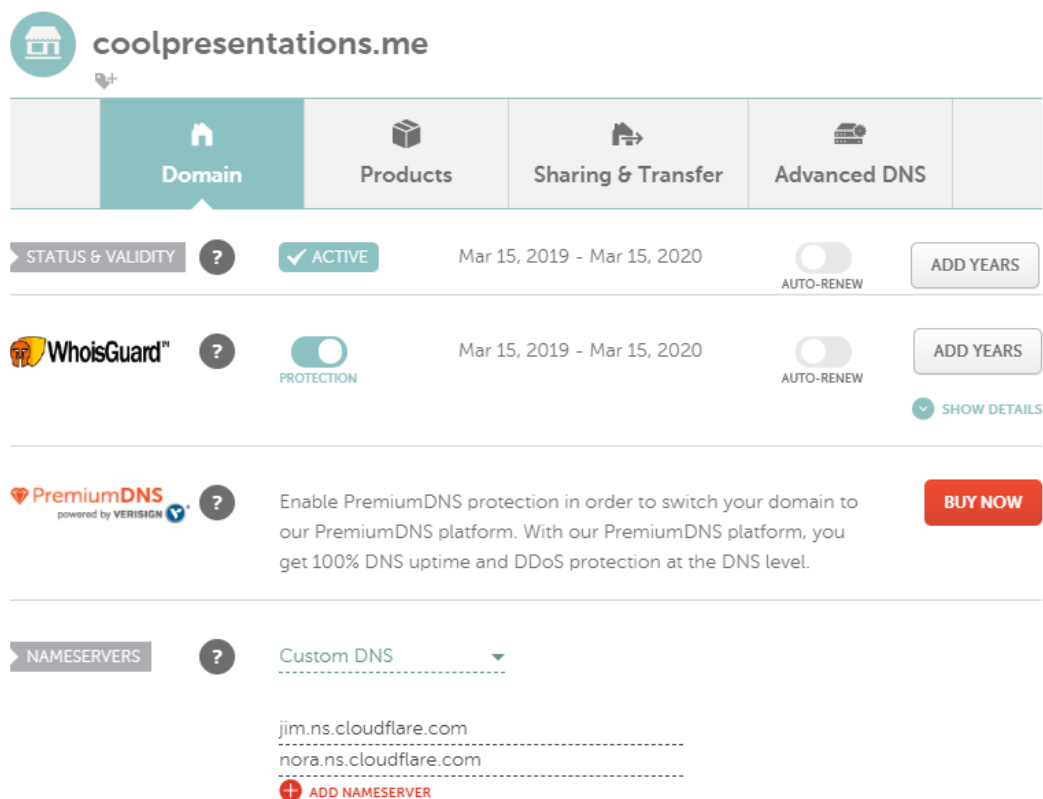


Figura 38: Recorte de la interfaz de NameCheap, delegación de nameservers

DNS

Manage your Domain Name System (DNS) settings


DNS Records

A, AAAA, and CNAME records can have their traffic routed through the Cloudflare system. Add more records using this form, and click the cloud next to each record to toggle Cloudflare on or off.

A

⬆

⬆





Type	Name	Value	TTL	Status	
A	coolpresent...	points to 188.166.1...	Automatic		<input type="button" value="X"/>
CNAME	www	is an alias of coolpr...	Automatic		<input type="button" value="X"/>

Figura 39: Recorte de la interfaz de DNS de Cloudflare

9 Gestión económica y sostenibilidad

Las siguientes secciones pretender describir el análisis de la gestión económica y de la sostenibilidad de nuestro proyecto. Para ello, primeramente presentaremos una pequeña autorreflexión sobre el compromiso personal con la sostenibilidad. Seguidamente continuaremos con el análisis de la gestión económica para acabar tratando el análisis de la sostenibilidad en los ámbitos económico, ambiental y social para nuestro proyecto.

9.1 Autoevaluación competencia sostenibilidad

Contestar a la encuesta: goo.gl/kWLMLE no ha sido una tarea fácil. Primeramente, desconocía parte de la terminología usada y además no era capaz de evaluarme, puesto que desconocía el estado del arte en el tema y no sabía qué niveles de conocimiento había por parte de los expertos. Esto me llevó a darme cuenta que estaba muy verde y debía empezar de inmediato a involucrarme mucho más en el tema de la sostenibilidad. Tenía un concepto erróneo de la sostenibilidad de las TIC, pues pensaba que el desarrollo Software no era importante para la contaminación del medioambiente, y mis reflexiones se limitaban a esto, a la contaminación del medioambiente.

Después de hacer una pequeña investigación sobre el significado y estado de la sostenibilidad de las TIC me decidí a contestar la encuesta con sinceridad. La gran parte de las preguntas fueron puntuadas con un 1 debido mi desconocimiento en el tema, las que mejores fueron puntuadas hacen referencia al análisis económico del proyecto. Esta encuesta me ha servido para darme cuenta que la sostenibilidad no se basa únicamente en la contaminación de lo que producimos, sino que, por ejemplo, también incluye el impacto que éste provoca en las costumbres de la sociedad.

En cuanto a mis puntos fuertes y débiles sobre la sostenibilidad, desgraciadamente debo admitir que son más los débiles que los fuertes. En los desarrollos de proyectos, tengo muy en cuenta el análisis económico y su viabilidad, pero no suelo empeñar mucho tiempo en el estudio de su impacto social o medioambiental. Agradezco que planteen esta encuesta ya que me ha permitido darme cuenta del bajo compromiso que he tenido con la sostenibilidad hasta ahora.

9.2 Dimensión económica - presupuesto

9.2.1 Costes directos

- Costes hardware:

Producto	Precio	Unidades	Vida útil	Amortización
Dell Latitutde 5580 15.6`` (incluye periféricos)	937€	1	5 años	76.70€

Total	937€			76.70€
--------------	-------------	--	--	---------------

Tabla 2: Estimación costes Hardware

- Costes software:

Producto	Precio	Unidades	Vida útil	Amortización
Windows 10	150€	1	5 años	25€
Visual Code	0€	1	-	-
BitBucket	0€	1	-	-
Google Chrome	0€	1	-	-
Safari	0€	1	-	-

Firefox	0€	1	-	-
Google Docs	0€	1	-	-
RoboTest	0€	1	-	-
Trello	0€	1	-	-
CDN CloudFlare	0€	1	-	-
Total	150€			25€

Tabla 3: Estimación costes Software

- **Costes recursos humanos:** Aunque la única persona que desarrolla el proyecto soy yo, dependiendo el rol que haga se contabiliza con un precio u otro.

Rol	Horas	€/hora	Salario
Project Manager	100	50	5.000
Desarrollador Software	400	35	14.000
Software Tester	100	20	2.000
Total			21.000

Tabla 4: Estimación costes recursos humanos

9.2.2 Costes indirectos

Producto	Precio	Unidades	Coste estimado
Hosting DigitalOc	10€/mes	4 meses	40€
Electricidad	0.12€/kWh	500 horas	60€
Fibra óptica	30€/mes	4 meses	120€

Costes oficina	120€	-	120€
Total			340€

Tabla 5: Estimación costes indirectos

9.2.3 Costes desviaciones e imprevistos “previstos”

En esta sección intentaremos identificar y estimar los costes generados por las desviaciones y aquellos que, aunque no se deberían de llegar a efectuarse realmente, debemos preverlos. Es por ello que los llamamos imprevistos “previstos”

Nuestra principal desviación, como comentamos, aparece en la resolución de bugs. Para ello se estima un presupuesto margen para cubrir los gastos de recursos humanos. En este caso se estima un 5% de los gastos humanos que equivalen a 1050€ los cuales equivalen a 30 horas como desarrollador software. Otra posible desviación o imprevisto que puede ser cuantificado es el caso que el equipo con el que desarrollamos se estropee. Para ello se estiman unos 150€ para el alquiler de un nuevo equipo durante los 4 meses de desarrollo del proyecto. Para acabar, estimaremos un 5% más al presupuesto final como contingencia por si llegasen imprevistos reales.

9.2.4 Control de costes

Debido al tipo de proyecto, el control de costes no tiene sentido en cuanto a costes de Hardware o Software, ya que son bastante pequeños en comparación a los costes destinados a recursos humanos. Es por ello que el control del presupuesto se centra en las desviaciones de costes producidos por los recursos humanos.

El presupuesto se revisará y ajustará cada vez que se haya terminado una tarea, puesto que sabremos ya exactamente el número de horas que se han llevado a cabo y las que aún nos quedan. De esta manera compararemos el número de horas realizadas con las estimadas y recalcularemos el presupuesto según las desviaciones. La fórmula a seguir será la siguiente:

$$\begin{aligned} \text{Desviación coste} &= (\text{CE} - \text{CR}) \cdot \text{HR} & \text{HE} &= \text{Horas estimadas} \\ \text{Desviación por consumo} &= (\text{HE} - \text{HR}) \cdot \text{CE} & \text{CE} &= \text{Coste estimado} \\ & & \text{HR} &= \text{Horas reales} \\ & & \text{CR} &= \text{Coste Real} \end{aligned}$$

9.2.5 Costes totales

En esta sección se agruparán todos los costes mencionados en las secciones anteriores. La siguiente tabla 6 muestra el resumen. Notar que se ha añadido el 5% de contingencia que se ha comentado.

Concepto	Coste estimado (€)
Costes HW	937
Costes SW	150
Costes RRHH	21.000
Costes indirectos	340
Costes por desviaciones	830

Subtotal	23.257
Contingencia 5%	1.200

Total	24.457
--------------	---------------

Tabla 6: Estimación costes totales

Si deseamos saber el coste estimado para cada tarea identificada en el Gantt simplemente debemos multiplicar el número de horas de la tarea por el precio medio según el rol. Por ejemplo para la tarea “Encuestas en tiempo real” que se han estimado 60 horas podemos estimar: $(50 \text{ horas} \times 35\text{€/h desarrollador}) + (10 \text{ horas} \times 50\text{€/hora project manager}) = 2.250\text{€}$.

9.2.6 Reflexión sostenibilidad económica

El plan detallado de los costes económicos de la realización del proyecto ya se ha estimado en las secciones anteriores. No lo considero un presupuesto elevado para las ventajas que propone. Actualmente, los softwares similares mencionados en el estado del arte, son privativos y requieren de pagos para añadir funcionalidades similares a las que este proyecto ofrece de manera gratuita. Recordar que este proyecto es y será de código libre por lo que toda la comunidad podrá usarlo sin ningún coste. Esto presenta una gran ventaja respecto a la competencia, puesto que los usuarios podrán modificar y añadir funcionalidades de manera gratuita según sus necesidades, y no solo eso, sino que también se beneficiarán de forma gratuita de los aportes de toda la comunidad.

9.3 Dimensión ambiental

Los recursos que se utilizarán para el desarrollo del proyecto ya se han comentado en el apartado “5.2.1 Costes directos” y en el “5.2.2 Costes indirectos”. Debido a que los únicos recursos son un ordenador portátil y la luz eléctrica que éste consume veo difícil mejorar el impacto medioambiental. Algunos de los aspectos que podríamos tener en cuenta vienen relacionados con el ahorro de energía en el uso de ordenadores. Por ejemplo, bajar el brillo de la pantalla cuando estemos en zonas oscuras, o encender el modo ahorro si no estamos haciendo un uso intensivo de CPU.

Se ha planteado también el reutilizar recursos, pero materialmente sólo utilizamos un ordenador portátil, el cual, ya es de segunda mano, y en caso de que se estropease, no compraríamos otro, sino que, como hemos comentado en el apartado de costes imprevistos, lo alquilaríamos. Para acabar con esta sección me gustaría aclarar que los Software que forman la competencia al proyecto no presentan ninguna ventaja en cuanto al impacto medioambiental, sino todo al contrario, al ser software privativo y no poder usarse libremente, obliga a tener que desarrollar otros proyectos de código libre desde cero que tengan las mismas características.

9.4 Dimensión social

El gran impacto social de este proyecto es a nivel personal. Es la primera vez que llevo a cabo un proyecto de tanta importancia y sin duda alguna será la forma de poder demostrar lo aprendido durante la carrera universitaria. Espero que este proyecto pueda darme la retroalimentación necesaria para saber en qué partes del desarrollo de un proyecto estoy más flojo, para poder mejorar en estos aspectos.

Podría ocurrir otro gran impacto social si este proyecto se llevase a las escuelas del país. Como sabemos muchas universidades y escuelas utilizan las clásicas transparencias para dar las clases, y el cambio hacia nuestra herramienta podría provocar un cambio radical en la forma de impartir las clases. Esto afectaría directamente tanto a profesores como a estudiantes. El cambio que comentamos consideramos que es un cambio positivo y de mejora, puesto que aumenta el “feedback” entre alumno y profesor y ayuda a los estudiantes a seguir las transparencias con mayor facilidad. No solo en las escuelas, como sabemos, esta herramienta es aplicable a muchos otros entornos, como en el mundo laboral, pero todos los impactos sociales que observamos son positivos, puesto que mejoran la comunicación y retroalimentación.

Me gustaría destacar aquí la importancia de que el proyecto sea de código libre, esto aporta muchos beneficios sociales en contraposición al código privativo:

- 1) Se puede ejecutar la aplicación cuando se quiera y para lo que se quiera.
- 2) Se puede aprender del código ya escrito .
- 3) Se puede mejorar.
- 4) Se puede modificar para adaptar a nuestras necesidades.
- 5) Se puede tomar como base para desarrollar otro software sin tener que empezar de cero, con la obligación de redistribuirlo (en el apartado de leyes hablamos de ello).

El efecto final de todos estos beneficios es la formación de grandes comunidades alrededor de proyectos de software libre.

10 Integración de conocimientos

Para identificar los conocimientos que se han puesto en práctica, haremos uso de las competencias que identificamos en el hito inicial:

- **CTI2.1:** Dirigir, planificar i coordinar la gestió de la infraestructura informàtica: hardware, software, xarxes i comunicacions. [Un poco]
- **CTI2.2:** Administrar i mantenir aplicacions, sistemes informàtics i xarxes de computadors (els nivells de coneixement i de comprensió són a les competències tècniques comunes). [Un poco]

Para alojar el portal web he montado un servidor en DigitalOcean. Además hemos configurado un dominio apuntando a este servidor junto con un certificado SSL para evitar ataques de “man in the middle”. Una vez tenía el servidor hay que configurarlo. El servidor lo he montado con la imagen base de Ubuntu-18-04. He instalado y configurado el software necesario, PHP, MySQL y Apache2.

Para la monitorización del servidor he utilizado “Node Exporter” de Prometheus, el cual publica métricas del servidor en el puerto 9100 en el path /metrics. Con estas métricas, las cuales se pintan en gráficas en Grafana podemos saber en todo momento el estado actual del servidor en cuanto a recursos de CPU, disco, memoria.. Por ejemplo, se ha tenido que instalar el programa “logrotate” el cual elimina y sube los logs de las aplicaciones (sobre todo Apache2 ya que genera un gran volumen de logs) a donde se configure ya que el disco de la máquina se llenaba.

- **CTI2.3:** Demostrar comprensió, aplicar i gestionar la garantia i la seguretat dels sistemes informàtics (CEIC6). [En profunditat]

Durante el desarrollo del proyecto se han tenido en cuenta muchos aspectos de la seguridad aprendidos durante la especialidad. Por ejemplo, todas las contraseñas, nombres de usuarios, mails, nombre de presentaciones... son encriptadas con SHA [48] Secure Hash Algorithm antes de enviarse a la base de datos. Otro ejemplo sería el control de los datos de entrada para evitar “SQL Injection” [49]

Además, la configuración del servidor, apache2, evita los ataques más conocidos como “SQL Injection”. No olvidar que se ha configurado un certificado SSL para evitar ataques “man in the middle”.

- **CTI3.1:** Concebre sistemes, aplicacions i serveis basats en tecnologies de xarxa, tenint en compte Internet, web, comerç electrònic, multimèdia, serveis interactius i computació ubiqua. [En profunditat]
- **CTI4:** Emprar metodologies centrades en l'usuari i l'organització per al desenvolupament, l'avaluació i la gestió d'aplicacions i sistemes basats en tecnologies de la informació que assegurin l'accessibilitat, l'ergonomia i la usabilitat dels sistemes. [Bastante]

El proyecto es en sí un servicio web online. Como todo servicio online se ha procurado que tenga una gran accesibilidad. Para ello, a parte de localizar el servidor en un punto céntrico, Amsterdam, se ha contratado una CDN CloudFlare para distribuir contenido y que tenga mejor accesibilidad. En todo momento se ha intentado conseguir la mejor usabilidad para todo tipo de personas y dispositivos, puesto que uno de los principales objetivos del proyecto es que el servicio sea usable, ligero y que se adapte al mayor número de pantallas posibles.

11 Identificación de leyes y regulaciones

Como este proyecto se basa en la creación de un servicio web online, deberemos cumplir con las leyes y regulaciones que este tipo de servicios presentan. Para empezar, en el portal deberá aparecer de forma visible la siguiente información:

- 1) Datos legales alusivos a la empresa, como nombre social de la compañía, datos de inscripción en el registro mercantil...
- 2) Avisos legales, en todo momento los usuarios deben poder visualizar los términos que regulan la utilización de la web.
- 3) Aviso de cookies, el usuario deberá aceptar la política de cookies y deberá saber para qué se utilizaran estos cookies.

Otra de las leyes que deberemos cumplir es la RGPD, ley general de protección de datos. Debido a que en el portal los usuarios se pueden registrar e introducir datos personales debemos cumplir con la RGPD estrictamente. Para ello deberemos cumplir las exigencias del siguiente documento:

<https://www.boe.es/doue/2016/119/L00001-00088.pdf>

Alguna de las características principales de esta normativa son:

- 1) Tener una política de tratamiento de datos. En ella se debe especificar quién está detrás de la página, para que se recopilan los datos, cómo los trata y qué opciones tiene el usuario para modificar o pedir la eliminación de dicha información.
- 2) Confirmación en formularios. Todo formulario debe tener un checkbox, por defecto no seleccionado, que confirme que hemos leído y aceptado la política de tratamiento de datos.

- 3) Seguridad de los datos. Debemos garantizar un mínimo en la seguridad de los datos que almacenamos, como por ejemplo, bases de datos encriptadas, formularios anti fuerza bruta, sistemas cortafuegos... Además debemos proporcionar mecanismos para que el usuario pueda borrar o trasladar sus datos a otra plataforma.
- 4) Gestión de cookies. Si hacemos uso de cookies debemos poner el famoso popup de confirmación de uso. El usuario debe poder identificar qué uso se le dará a las cookies y debe poder cancelarlo si lo desea.

En la siguiente figura 40 podemos observar el aviso de cookies que muestra la aplicación cuando accedemos por primera vez.

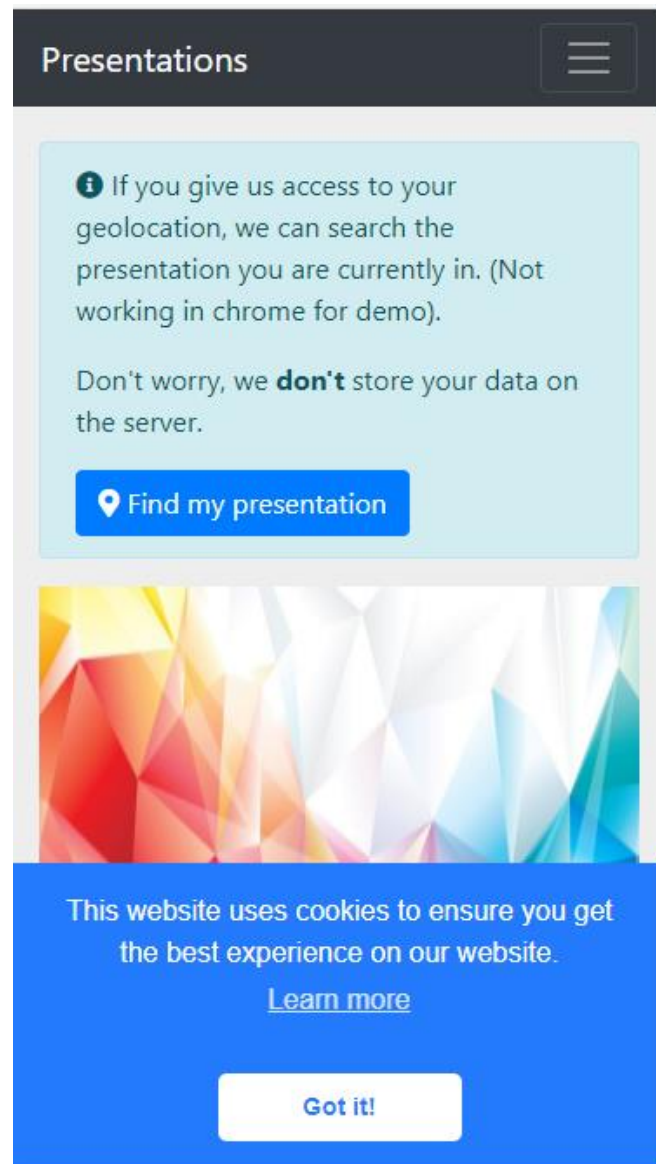


Figura 40: Recorte de la interfaz donde observamos el aviso de uso de cookies

En cuanto a las normativas que aplican a proyecto en sí, es decir, al código, he decidido que el proyecto sea de código libre bajo la licencia Affero GPLv3 [50]. Con esto pretendo que el proyecto no pueda privatizarse y sea siempre libre para toda la comunidad. La idea, desde el inicio, fue que la aplicación pudiera ser explotada libremente por centros educativos. La principal diferencia entre la licencia AfferoGPLv3 (AGPL) y la licencia GPLv3 [51] es que obliga a aquellos que hayan modificado código a redistribuirlo si es usado a través de la red, es decir, si crean una web utilizando el código están obligados a distribuirlo en beneficio de toda la

comunidad. Esta restricción es muy útil si hemos programado un CMS o una aplicación web como en este caso.

12 Conclusiones y reflexiones finales

Como conclusión principal me gustaría comentar que el proyecto ha sido capaz de resolver la problemática definida, consiguiendo una aplicación funcional que ayuda a lograr una comunicación más eficiente en las presentaciones que usan diapositivas como soporte. Comentar también que se espera que este proyecto sirva como base para futuras iteraciones de desarrollo, puesto que es de código abierto, y que con el tiempo se añadan más funcionalidades a la aplicación.

Como conclusiones secundarias, o más tecnológicas, me gustaría comentar algunos puntos:

- Bootstrap me ha parecido una librería asombrosa, y me ha facilitado muchísimo el tener una aplicación responsive que se adapte a todo tipo de pantallas.
- Javascript es un lenguaje muy potente que tiene multitud de usos, en mi caso me ha sorprendido la facilidad para gestionar el DOM de HTML y su rendimiento, puesto que el 50% aproximadamente de mi código está en JS y sin ser un experto la aplicación es completamente funcional.
- Sin los aportes de la comunidad informática, es decir, de las librerías y tecnologías de código abierto, este proyecto no podría haberse realizado en el tiempo marcado ni con los costes planificados.
- Los servicios privados “gratuitos”, como el uso de CloudFlare, están muy bien para empezar con desarrollos pequeños.
- Un stack LAMP, como el del proyecto, es muy fácil de implementar y su utilidad es muy amplia.

Personalmente, me ha gustado muchísimo la realización del proyecto, y tengo pensado continuar una vez acabado el TFG. He hecho uso de multitudes de lenguajes y tecnologías muy diversas, que aunque sabía lo que hacían, no me había peleado con muchas de ellas aún, por ejemplo con bootstrap o los droplets de DigitalOcean. Sinceramente, el desarrollo del proyecto, me ha ayudado mucho a adquirir confianza para diseñar mis propias soluciones sin depender de nadie, siempre intentando utilizar tecnologías punteras. Otra de las aptitudes que he conseguido mejorar es la del aprendizaje autónomo, puesto que me he tenido que buscar la vida en numerosas ocasiones para llegar a la mejor solución posible sin tener un conocimiento profundo de la materia.

12.1 Mejoras de futuro

El proyecto se puede ampliar en muchos aspectos pero me gustaría destacar algunas funcionalidades que creo que mejorarían la aplicación en gran medida. No he implementado estas funcionalidades debido a que tenía que acotarlas para conseguir un proyecto de alrededor de unas 500 horas.

- Encuestas de feedback personalizables.
- Posibilidad de poder añadir comentarios de texto cómo feedback.
- Manual de usuario interactivo.
- Recordatorio de dudas con imagen de la página en la que se anotó la duda y enlace a ella.
- Ranking de usuarios con más encuestas respondidas correctamente.
- Pasar de página en los dispositivos móviles a la vez que el presentador
- Ranking de presentaciones más populares

Como he comentado hay otros aspectos que podrían mejorar. Podrían implementarse websockets para mostrar los resultados votados en las encuestas en tiempo real, en vez de hacer peticiones ajax periódicamente. También se podría aplantillar el servidor con las configuraciones necesarias, haciendo uso de Terraform [52] por ejemplo, para poder replicarlo fácilmente si fuera necesario. Otra posible mejora sería levantar otro droplet para el stack de monitorización. También podríamos mejorar la documentación del código para favorecer aportes de la comunidad. Aunque hay muchas mejoras que podrían llevarse a cabo, creo que como primera iteración y prueba de concepto ha sido bastante buena para ver que la aplicación tiene potencial.

13 Bibliografía

- [1] Microsoft Powerpoint. (n.d.). Retrieved September 20, 2018, from <https://www.office.com/launch/powerpoint>
- [2] Kahoot! (n.d.). Retrieved September 20, 2018, from <https://kahoot.it/>
- [3] Live interactive audience participation. (2018, June 10). Retrieved September 20, 2018, from <https://www.poll Everywhere.com/>
- [4] UPC, D. L. (2014). Reflexiones sobre el uso de transparencias en clase. *Arquitectura De Computadors*, 1-8. Retrieved September 20, 2018. David López.
- [5] Kanban. (2018, August 30). Retrieved September 20, 2018, from <https://en.wikipedia.org/wiki/Kanban>
- [6] Scrum (software development). (2018, October 14). Retrieved September 20, 2018, from [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))
- [7] Windows 10. (2018, September 20). Retrieved September 21, 2018, from <https://www.microsoft.com/es-es/software-download/windows10>
- [8] Visual Studio Code - Code Editing. Redefined. (2016, April 14). Retrieved September 21, 2018, from <https://code.visualstudio.com/>
- [9] Representational state transfer. (2018, September 25). Retrieved September 21, 2018, from https://en.wikipedia.org/wiki/Representational_state_transfer
- [10] MySQL. (2018, September 10). Retrieved September 22, 2018, from <https://www.mysql.com/>
- [11] Git. (2018, October 14). Retrieved September 22, 2018, from <https://en.wikipedia.org/wiki/Git>
- [12] Bitbucket | The Git solution for professional teams. (n.d.). Retrieved September 22, 2018, from <https://bitbucket.org/>
- [13] Google Hangout. (n.d.). Retrieved September 22, 2018, from <https://hangouts.google.com/>
- [14] Trello. (n.d.). Retrieved September 22, 2019 from <https://trello.com>

- [15] C. (n.d.). CastInfo/robotest. Retrieved September 23, 2018, from <https://github.com/CastInfo/robotest>
- [16] DigitalOcean | Cloud Computing, Simplicity at Scale (n.d.). Retrieved January 24, 2019, from <https://www.digitalocean.com/>
- [17] Cloudflare, The Web Performance & Security Company. | Cloudflare (n.d.). Retrieved January 28, 2019, from <https://www.cloudflare.com/>
- [18] Google Chrome. (n.d.). Retrieved September 25, 2018, from <https://www.google.com/chrome/>
- [19] Safari. (n.d.). Retrieved September 25, 2018, from <https://www.apple.com/lae/safari/>
- [20] The new, fast browser for Mac, PC and Linux | Firefox. (n.d.). Retrieved September 25, 2018, from <https://www.mozilla.org/en-US/firefox/>
- [21] NASA. (n.d.). Retrieved February 15, 2019, from <https://www.nasa.gov>
- [22] LESS “El lenguaje dinámico de hojas de estilo”. (n.d.). Retrieved February 15, 2019, from <http://amatellanes.github.io/lesscss.org/>
- [23] PDF. (2019, February 5). Retrieved February 15, 2019, from <https://es.wikipedia.org/wiki/PDF>
- [24] Mozilla. (2019, February 5). Retrieved February 15, 2019, from https://es.wikipedia.org/wiki/Fundaci%C3%B3n_Mozilla
- [25] phpMyAdmin. (n.d.). Retrieved March 1, 2019, from <https://www.phpmyadmin.net/>
- [26] tablas rainbow | tablas arcoíris. (2019, January 5). Retrieved March 1, 2019, from https://es.wikipedia.org/wiki/Tabla_arco%C3%ADris
- [27] WebSockets | MDN. (n.d.). Retrieved March 1, 2019, from <https://developer.mozilla.org/es/docs/WebSockets-840092-dup>
- [28] Transmission Control Protocol. (2018, August 18). Retrieved September 20, 2018, from https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [29] Polling. (2019, October 23). Retrieved March 1, 2019, from <https://es.wikipedia.org/wiki/Polling>

- [30] AWS | Cloud Computing - Servicios de informática en la nube. (n.d). Retrieved March 2, 2019, from <https://aws.amazon.com/es/>
- [31] Google Cloud: Cloud Computing Services. (n.d). Retrieved March 2, 2019, from <https://cloud.google.com/>
- [32] Heroku: Cloud Application Platform. (n.d). Retrieved March 2, 2019, from <https://www.heroku.com/>
- [33] Ubuntu 18.04.02 LTS (Bionic Beaver). (n.d). Retrieved March 2, 2019, from <http://releases.ubuntu.com/18.04/>
- [34] Secure Shell. (2018, December 26). Retrieved March 3, 2019, from https://es.wikipedia.org/wiki/Secure_Shell
- [35] Secure Copy. (2018, September 13). Retrieved March 3, 2019, from https://es.wikipedia.org/wiki/Secure_Copy
- [36] Jenkins. (n.d). Retrieved March 3, 2019, from <https://jenkins.io/>
- [37] LAMP. (2018, August 19). Retrieved March 4, 2019, from <https://es.wikipedia.org/wiki/LAMP>
- [38] LogRotate - Linux man page. (n.d). Retrieved February 14, 2019 from <https://linux.die.net/man/8/logrotate>
- [39] Unix. (2019, February 19). Retrieved March 4, 2019, from <https://es.wikipedia.org/wiki/Unix>
- [40] Directory Listing - Httpd Wiki. (n.d). Retrieved March 4, 2019, from <https://wiki.apache.org/httpd/DirectoryListings>
- [41] Prometheus-Monitoring system & time series database. (n.d). Retrieved March 7, 2019, from <https://prometheus.io/>

- [42] Alertmanager | Prometheus. (n.d). Retrieved March 7, 2019, from <https://prometheus.io/docs/alerting/alertmanager/>
- [43] Prometheus/node-exporter: Exporter for machine metrics. (n.d). Retrieved March 7, 2019, from https://github.com/prometheus/node_exporter
- [44] Autoridad de certificación. (2019, February 12). Retrieved March 8, 2019, from https://es.wikipedia.org/wiki/Autoridad_de_certificaci%C3%B3n
- [45] Let's Encrypt - Free SSL/TLS Certificates. (n.d). Retrieved March 8, 2019, from <https://letsencrypt.org/>
- [46] Certbot. (n.d). Retrieved March 8, 2019, from <https://certbot.eff.org/>
- [47] NameCheap | Cheap Domains Name. (n.d). Retrieved January 26, 2019, from <https://www.namecheap.com/>
- [48] SHA - Secure Hash Algorithm. (2019, February 1). Retrieved February 15, 2019, from https://es.wikipedia.org/wiki/Secure_Hash_Algorithm
- [49] SQL Injection. (2019, February 10). Retrieved February 15, 2019, from https://en.wikipedia.org/wiki/Inyección_SQL
- [50] Licència Pública General Affero de GNU. (n.d). Retrieved February 15, 2019, from <https://www.gnu.org/licenses/agpl-3.0.html>
- [51] The GNU General Public License v3.0. (n.d). Retrieved February 15, 2019, from <https://www.gnu.org/licenses/gpl-3.0.en.html>
- [52] Terraform by HashiCorp. (n.d). Retrieved February 5, 2019, from <https://www.terraform.io/>

14 Índice de figuras

Figura 1: Recorte del tablón de Trello.....	17
Figura 2: Tiempo estimado por tarea, 1er Gantt.....	24
Figura 3: Tiempo estimado por tarea, 2º Gantt.....	26
Figura 4: Esquema funcionamiento AJAX	28
Figura 5: Pantallazo de una presentación en Iphone 6 con orientación horizontal.....	29
Figura 6: Pantallazo de una presentación en Iphone 6 con orientación horizontal, modal de dudas	29
Figura 7: Pantallazo de una presentación en Iphone 6 con orientación vertical, sin pantalla completa.....	30
Figura 8: Pantallazo de la portada de la aplicación en pc	31
Figura 9: Logos de las 7 librerías utilizadas	33
Figura 10: Vista navegador del frontal de phpMyAdmin	34
Figura 11: Recorte de la función “save_into_database”	35
Figura 12: Esquema de las tablas y campos de la base de datos	38
Figura 13: Sentencias para actualizar la base de datos de la versión 3 a la 4	38
Figura 14: Sentencias para crear la tabla ‘presentations’	39
Figura 15: Recorte de código que genera la apertura del modal de duas.....	41
Figura 16: Recorte de la función AJAX que implementa “polling”.....	43
Figura 17: Recorte del modal de resultados de encuestas	44
Figura 18: Recorte de código en el que aparecen plantillas HTML	45

Figura 19: Recorte de código en el que aparece la inserción de plantillas HTML	46
Figura 20: Recorte de código JS que realiza la conexión al servidor	47
Figura 21: Recorte de código JS que comprueba los datos introducidos.....	48
Figura 22: Recorte de código PHP de la función “check_name_and_pass”	49
Figura 23: Recorte de código PHP de la función “login”	50
Figura 24: Recorte de código en el que aparece la función “get_presentations_where”.	51
Figura 25: Recorte de código PHP en el que se implementa la capa de seguridad	52
Figura 26: Recorte del navegador en el que se observa una url con clave correcta.....	53
Figura 27: Recorte del navegador en el que se observa una url con clave incorrecta ..	53
Figura 28: Recorte de la interfaz de DigitalOcean en el que aparece las características del “Droplet”	55
Figura 29: Recorte de la configuración de Apache	57
Figura 30: Recorte de la interfaz de Alertas de DigitalOcean	58
Figura 31: Fichero configuración prometheus.yml	59
Figura 32: Fichero configuración droplet-alerts.yml	59
Figura 33: Recorte de algunas métricas publicadas por node_exporter	60
Figura 34: Recorte de la interfaz de CloudFlare	62
Figura 35: Recorte de la interfaz de CloudFlare	63
Figura 36: Recorte de la interfaz de Certbot	64
Figura 37: Recorte del certificado entregado por coolpresentations.me	65
Figura 38: Recorte de la interfaz de NameCheap, delegación de nameservers	66

Figura 39: Recorte de la interfaz DNS de CloudFlare67

Figura 40: Recorte de la interfaz donde observamos el aviso de uso de cookies78

15 Índice de tablas

Tabla 1: Tiempo estimado por tarea	23
Tabla 2: Estimación costes hardware	69
Tabla 3: Estimación costes software	70
Tabla 4: Estimación costes recursos humanos	70
Tabla 5: Estimación costes indirectos	71
Tabla 6: Estimación coste totales	72

16 Anexo I: Manual de usuario

Empezaremos mostrando la pàgina principal de la pàgina web para familiarizarnos con ella, figura 1:

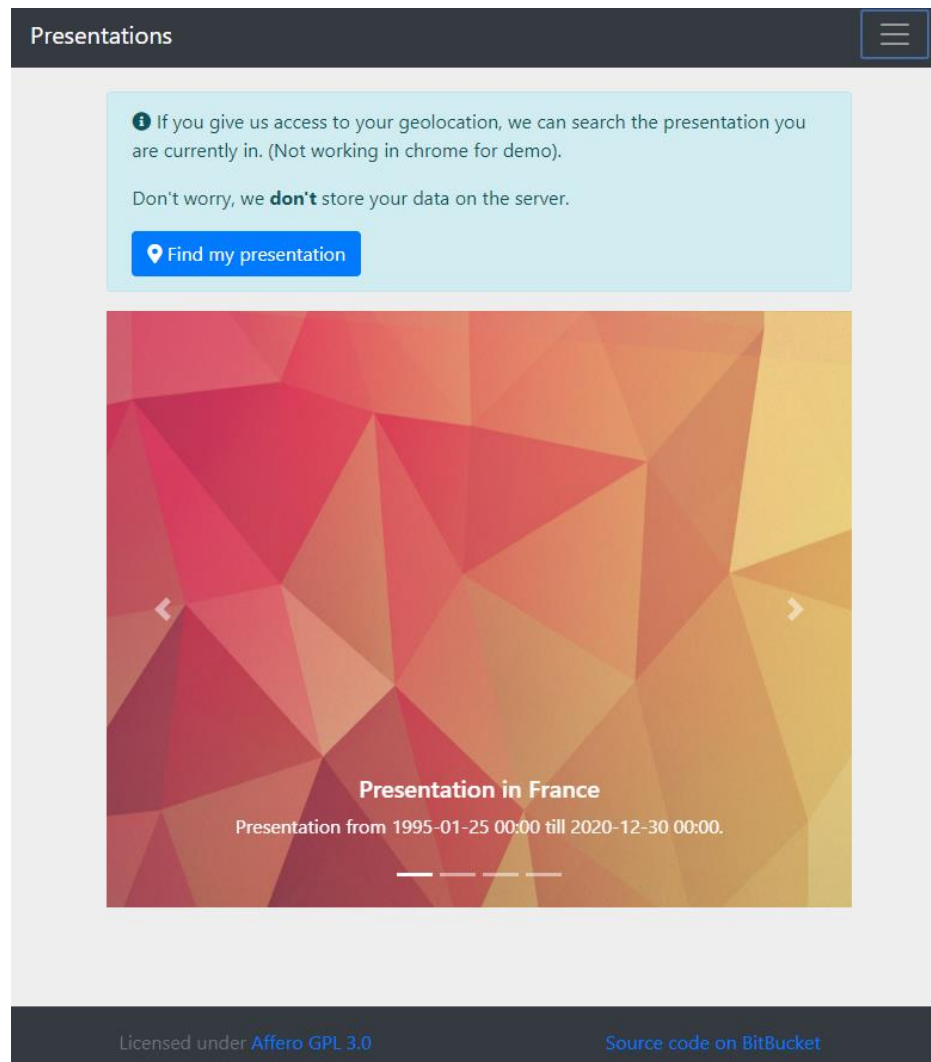


Figura 1: Recorte de la pàgina de inicio de la web.

Con el botón de la parte superior derecha, figura 2 señalada en azul, podemos desplegar la botonera.

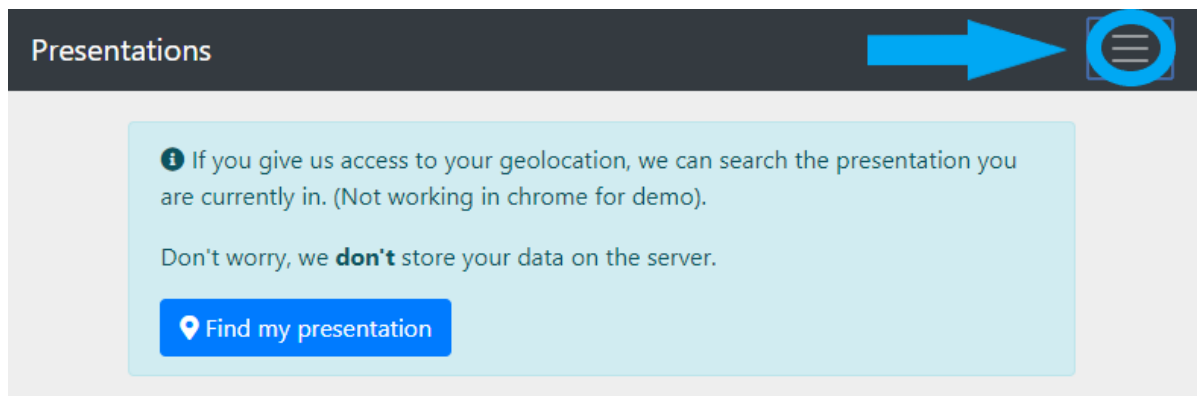


Figura 2: Recorte de la pàgina principal señalando la botonera

Una vez desplegada la botonera, nos aparecerá el botón de loguearse, en la figura 3 coloreado en verde. Apretando en él nos llevará a la página de registro/login tal y como muestra la figura 4.

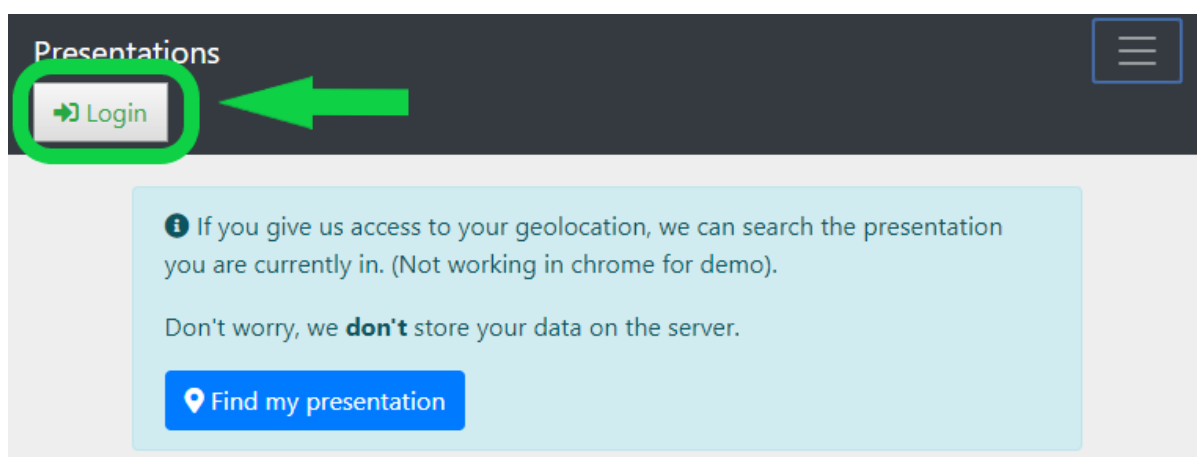
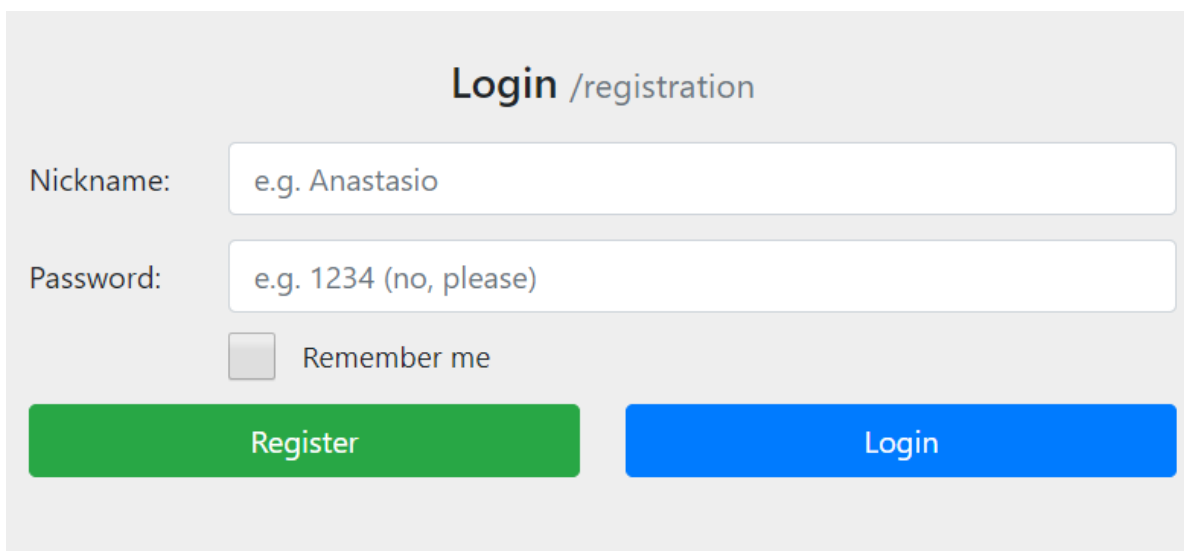


Figura 3: Recorte de la pàgina principal señalando el botón de registro.

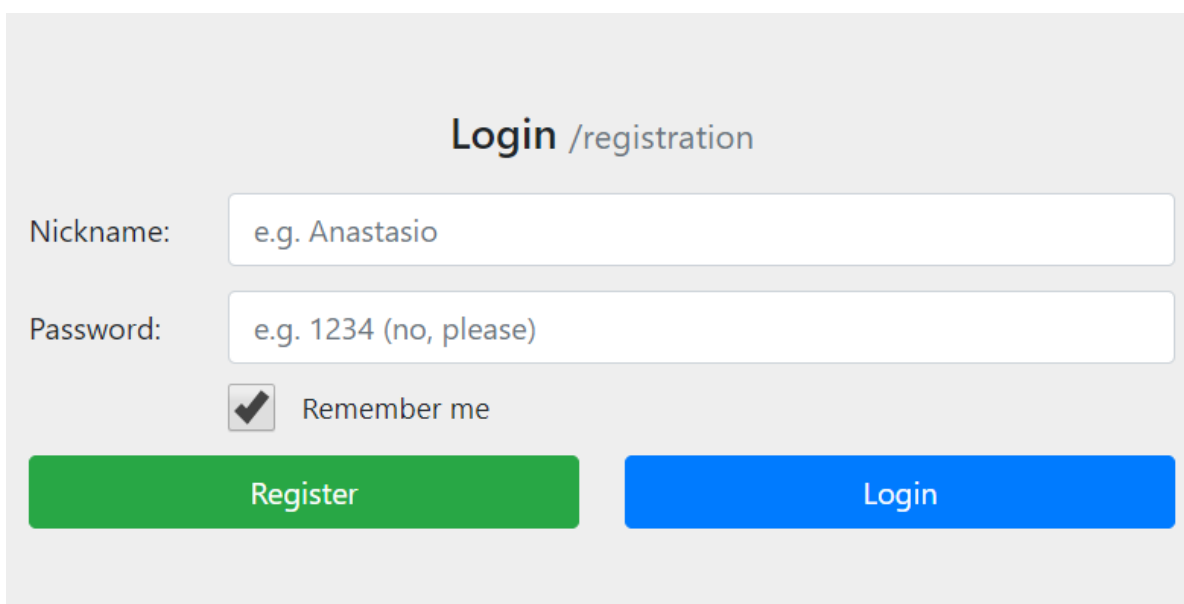
En la pàgina de registro/login tendremos que introducir un nombre de usuario y una contraseña que contenga mínimo 8 caràcteres. Una vez realizado este paso apretaremos el botón de registro coloreado en verde en la figura 4. En el caso de tener una cuenta registrada en la pàgina web, tendremos que apretar el botón de acceder, en la figura 4 coloreado en azul.



The image shows a web form titled "Login /registration". It contains two input fields: "Nickname:" with the placeholder text "e.g. Anastasio" and "Password:" with the placeholder text "e.g. 1234 (no, please)". Below the password field is a checkbox labeled "Remember me", which is currently unchecked. At the bottom of the form are two buttons: a green "Register" button and a blue "Login" button.

Figura 4: Recorte de la pàgina de registro y login.

Durante el registro, la pàgina nos permite seleccionar la opción de recordar, la cual activaremos apretando el checkbox que aparece. Sabremos qué està activado cuando nos aparece un “check” en el recuadro como se muestra en la figura 5.



This image is similar to Figure 4, showing the same "Login /registration" form. However, the "Remember me" checkbox is now checked, indicated by a small black checkmark inside the box. The "Register" and "Login" buttons remain at the bottom.

Figura 5: Recorte de la pàgina de registro y login.

Despuès de registrar-nos la web nos llevará a la pàgina principal de nuevo. Seleccionaremos el botón superior derecho para abrir el desplegable tal y como se

muestra en la figura 2. Esta vez tendremos dos opciones, crear nuevas presentaciones o acceder a nuestras presentaciones.

Empezaremos por explicar cómo se crea una nueva presentación. Lo primero será apretar el botón “new presentation” subrayado en naranja en la figura 6.



Figura 6: Recorte del desplegable con la opción de nueva presentación y mis presentaciones

El primer paso para subir una nueva presentación es apretar el botón “open the file Browser” o arrastrar el documento hasta el recuadro blanco donde pone “drag & drop files here”. El documento siempre tendrá que subirse en formato pdf y deberá pesar menos de 5 megas.

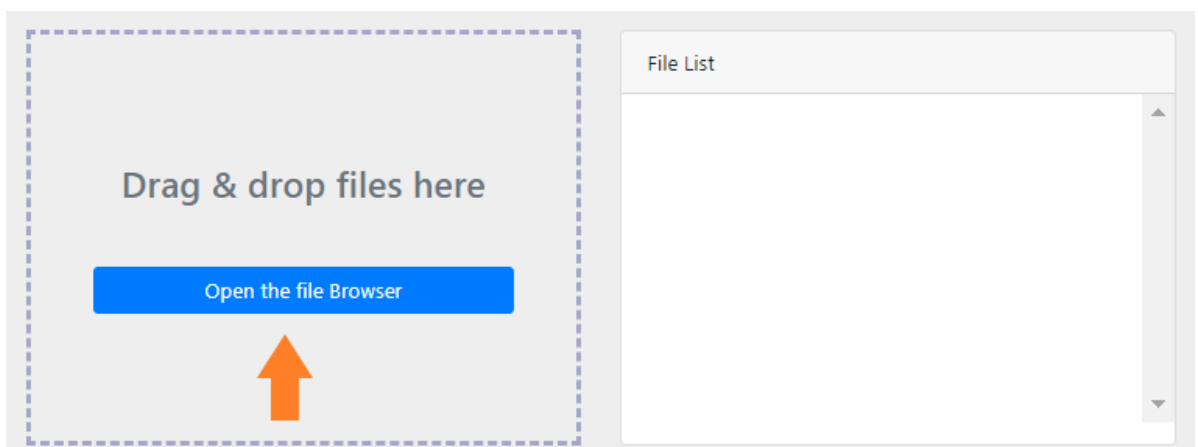


Figura 7: Recorte de la página para subir nuevas presentaciones

Una vez hemos seleccionado la presentación deseada nos aparecerà dicho documento en el recuadro de la derecha tal y como se muestra en la figura 8.

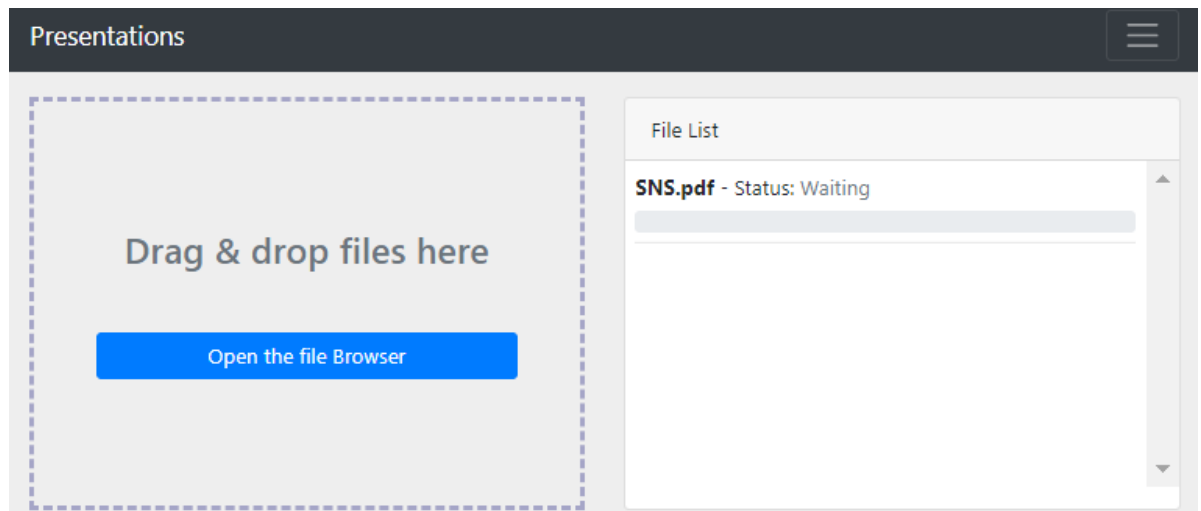


Figura 8: Recorte de la pàgina para subir nuevas presentaciones

A continuación, en el recuadro “presentation name” escribiremos el nombre de la presentación y en el recuadro “access code” escribiremos el código con el cual se podrá acceder a la presentación. Podemos dejarlo en blanco si deseamos que la presentación sea pública. Más abajo nos aparece un checkbox que si lo seleccionamos nos da la posibilidad de que los espectadores puedan descargar la presentación.

The image shows a form with three main sections. The first section is labeled "Presentation name:" and contains a text input field. The second section is labeled "Access code:" and contains another text input field. The third section contains a checkbox labeled "Downloadable?".

Figura 9: Recorte de la pàgina del nombre de la presentación y el código

Seguidamente, tenemos la opción de acotar el tiempo en el cual se va a poder visualizar el documento con día, mes, año y hora de inicio y de final.

Init time:

dd/mm/aaaa

--:--

End time:

dd/mm/aaaa

--:--

Figura 10: Recorte de la pàgina para poner fecha y hora a la presentación

Para añadir una localización a la presentación y que sólo se encuentre disponible para los usuarios que se encuentren en esa misma posición o 100 metros alrededor, apretaremos el botón azul “add location” y se nos abrirà el desplegable con un mapa.

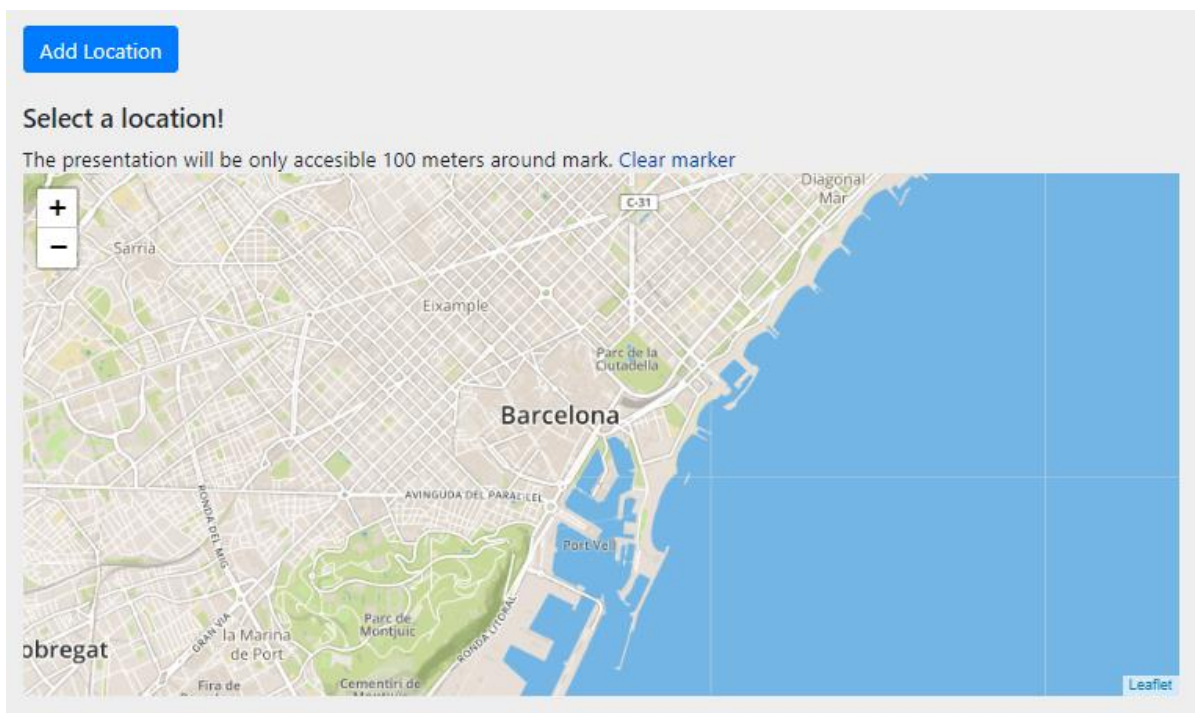


Figura 11: Recorte del mapa de localización

Para seleccionar la ubicación que deseamos, debemos apretar con el botón izquierdo del ratón el lugar del mapa donde queramos que se visualice el documento. En el caso que queramos cambiar la ubicación una vez marcada, apretaremos con el

botón izquierdo en otro punto del mapa. Para borrar la ubicación apretaremos el botón “clear marker” subrayado de color rojo en la figura 12.

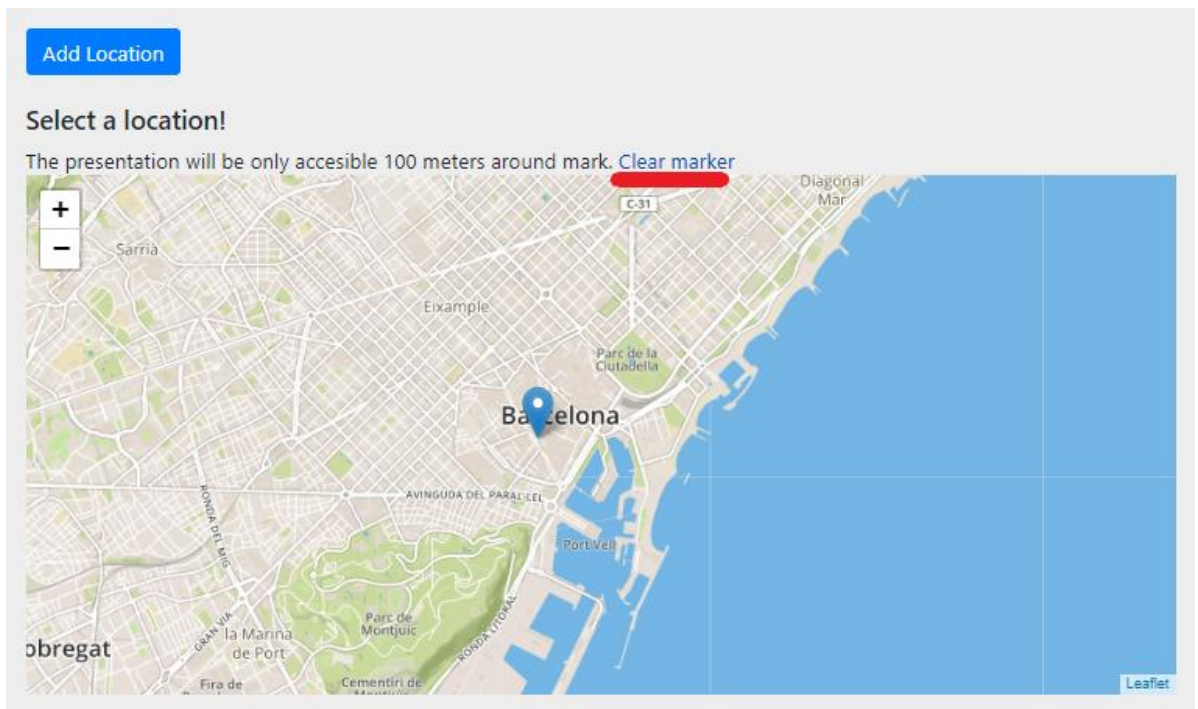


Figura 12: Recorte del mapa de localización

Otra de las opciones disponibles es el botón “add survey 1” que si lo apretamos nos permite añadir cuestionarios a la presentación. En el primer recuadro escribimos la pregunta, en el segundo recuadro escribimos el número de página dónde queremos que aparezca la cuestión. El recuadro con el nombre “position x and y” nos permite seleccionar en qué parte de la diapositiva queremos que se visualice la encuesta y por último el recuadro “width and height” define el tamaño y la altura de la encuesta. Al final de la página encontramos un checkbox que si lo marcamos tenemos la posibilidad que la pregunta formulada se puedan votar varias respuestas.

También tendremos que tener en cuenta que, cuando apretemos el botón “add survey 1”, se generará directament el botón “add survey 2” por si queremos realizar un segundo cuestionario.



The screenshot shows the 'Add Survey 1' form in the CoolPresentations application. It includes fields for 'Question', 'Page number', 'Position X and Y (px)' (with separate fields for X and Y), and 'Width and Height (px)' (with separate fields for width and height). There is a checkbox for 'Multiple choice allowed?' and an 'Add Answers' button at the bottom.

Figura 13: Recorte del generador de cuestionarios

Una vez formulada la pregunta, apretaremos el botón “add answers” para añadir las respuestas al cuestionario tal y como vemos en la figura 14.



The screenshot shows the 'Add Answers' form in the CoolPresentations application. It features an 'Add Answers' button at the top and five input fields labeled 'Answer1:', 'Answer2:', 'Answer3:', 'Answer4:', and 'Answer5:'. An 'Add Survey 2' button is located at the bottom.

Figura 14: Recorte de las respuestas del cuestionario

Para finalizar, nos aparecen dos botones. El de color verde con el nombre “submit” es para subir la presentación una vez hemos realizado los pasos anteriores. No obstante, si queremos borrar la presentación que íbamos a subir en un inicio, apretamos el botón rojo con el nombre “reset” que nos borrará sólo el documento.

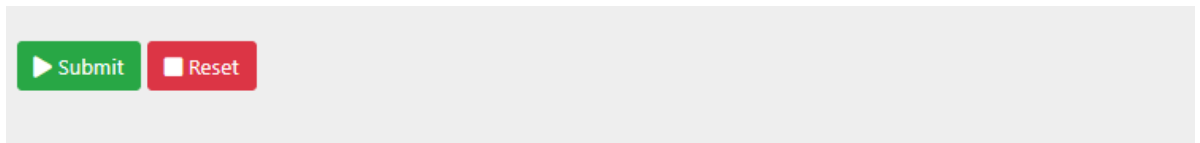


Figura 15: Recorte de las botoneras del cuestionario

Una vez hemos colgado la presentación, podemos acceder a ella en la página de inicio apretando encima de el documento. También podemos acceder a ella apretando el botón situado arriba la derecha que nos abrirá un desplegable con la opción “my presentations” subrayado en color azul en la figura 6.

Apretando el botón “my presentations” nos lleva a las presentaciones que tenemos colgadas. En esta página podemos ver las valoraciones de los espectadores que han visitado la presentación. También podemos visualizarla apretando encima. Por último, podemos borrar la presentación apretando el botón rojo “remove presentation”.

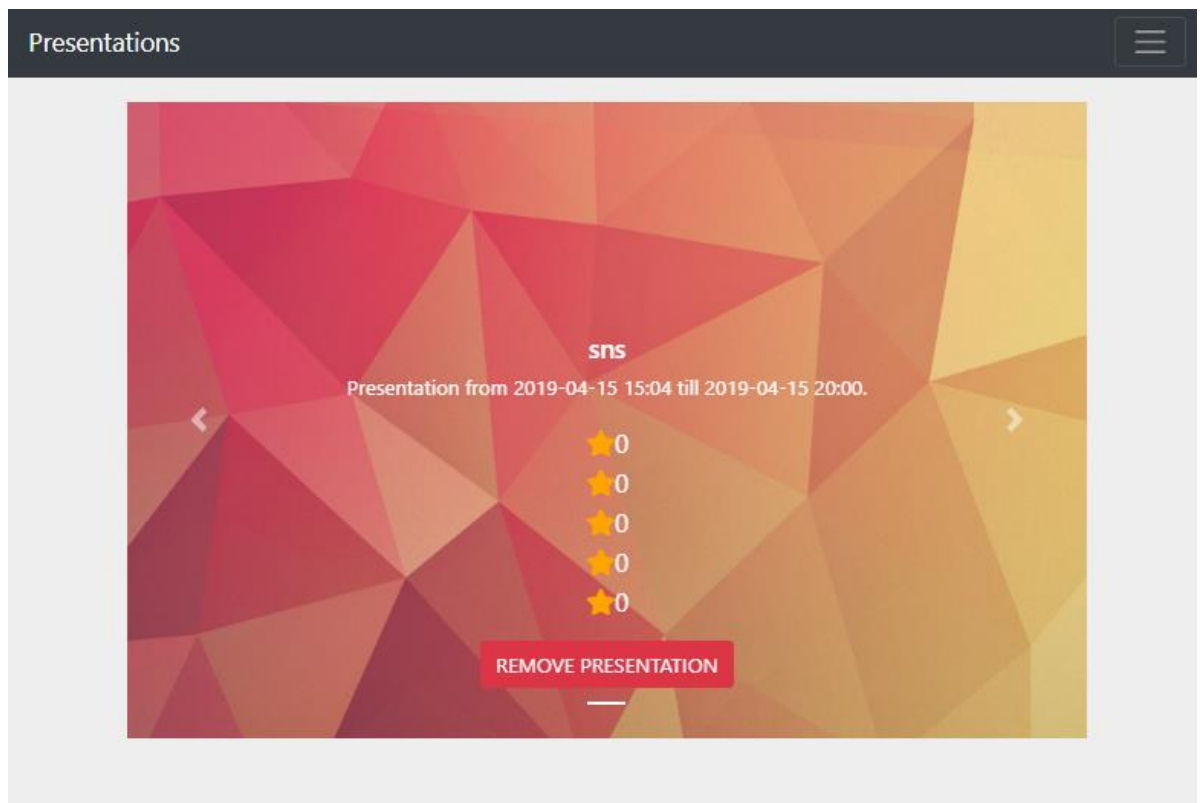


Figura 16: Recorte de la presentación en la sección “my presentations”

En el caso que hayamos puesto una contraseña a la presentación nos aparecerà el recuadro que podemos ver en la figura 17.

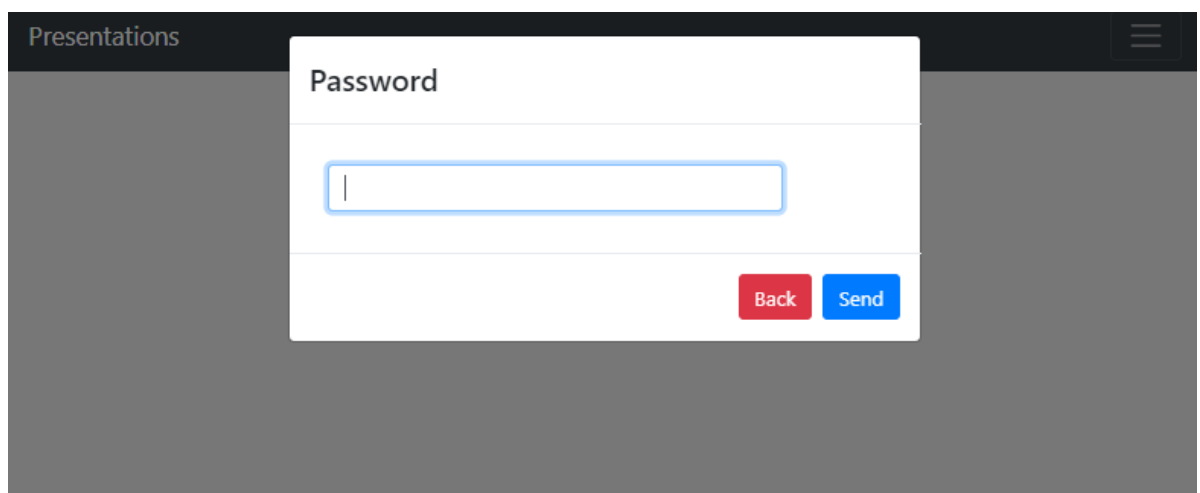


Figura 17: Recorte de la contraseña de la presentación.

Para pasar las páginas de la presentación podemos apretar las flechas que encontramos abajo de la presentación. En el caso que estemos con el ordenador, podemos hacerlos con las flechas del teclado, la de la derecha y abajo para ir hacia atrás y la de la derecha y arriba para ir hacia delante.

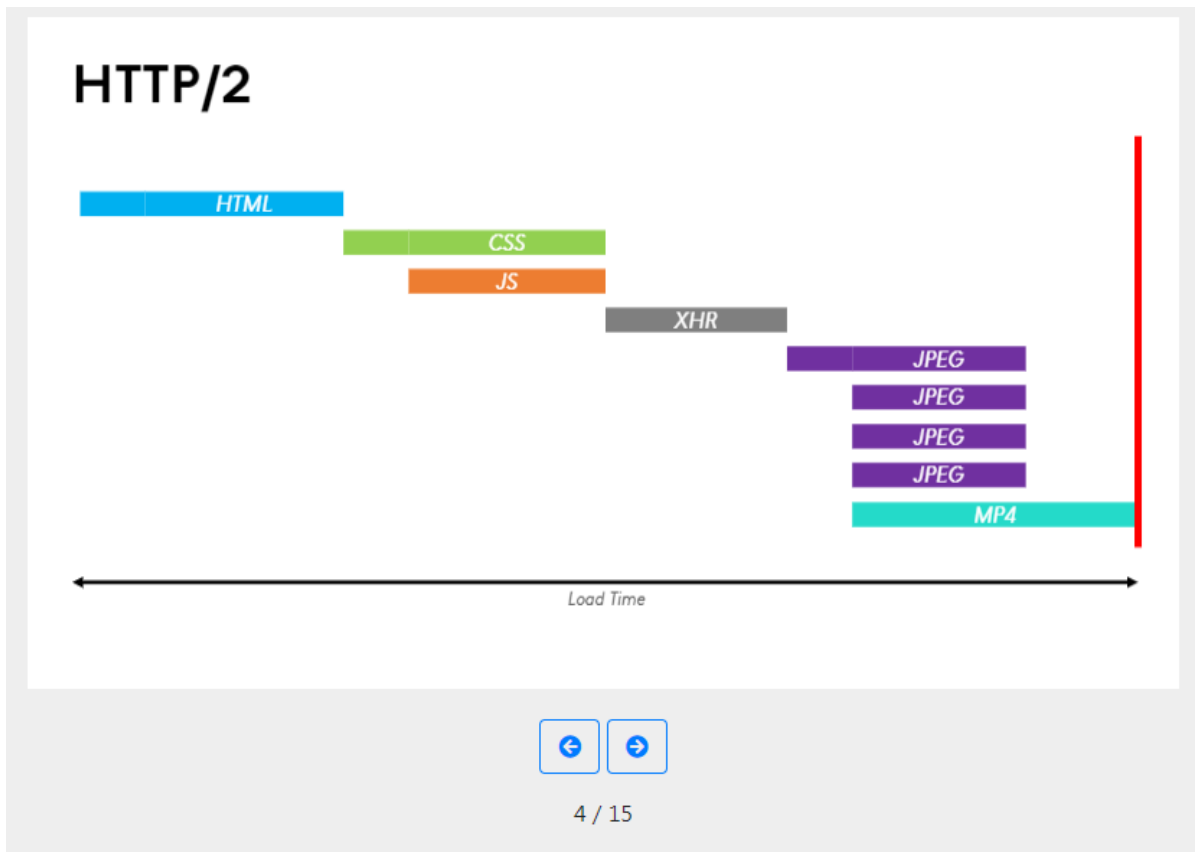


Figura 18: Recorte de la presentación

Otra de las funcionalidades de la web hace referencia al espectador. Si el usuario tiene una duda durante la presentación, deberá hacer un **círculo** con el cursor, o con el dedo, en la parte en la cual tiene la pregunta y esta será recordada al final de la presentación. En caso de querer borrar el dibujo simplemente se debe hacer doble-click sobre la pantalla y se eliminará el dibujo. Escribiremos la duda en el modal que aparece y apretaremos el botón azul “save”. Si el círculo no se cierra no se abrirá la pestaña para generar la pregunta.

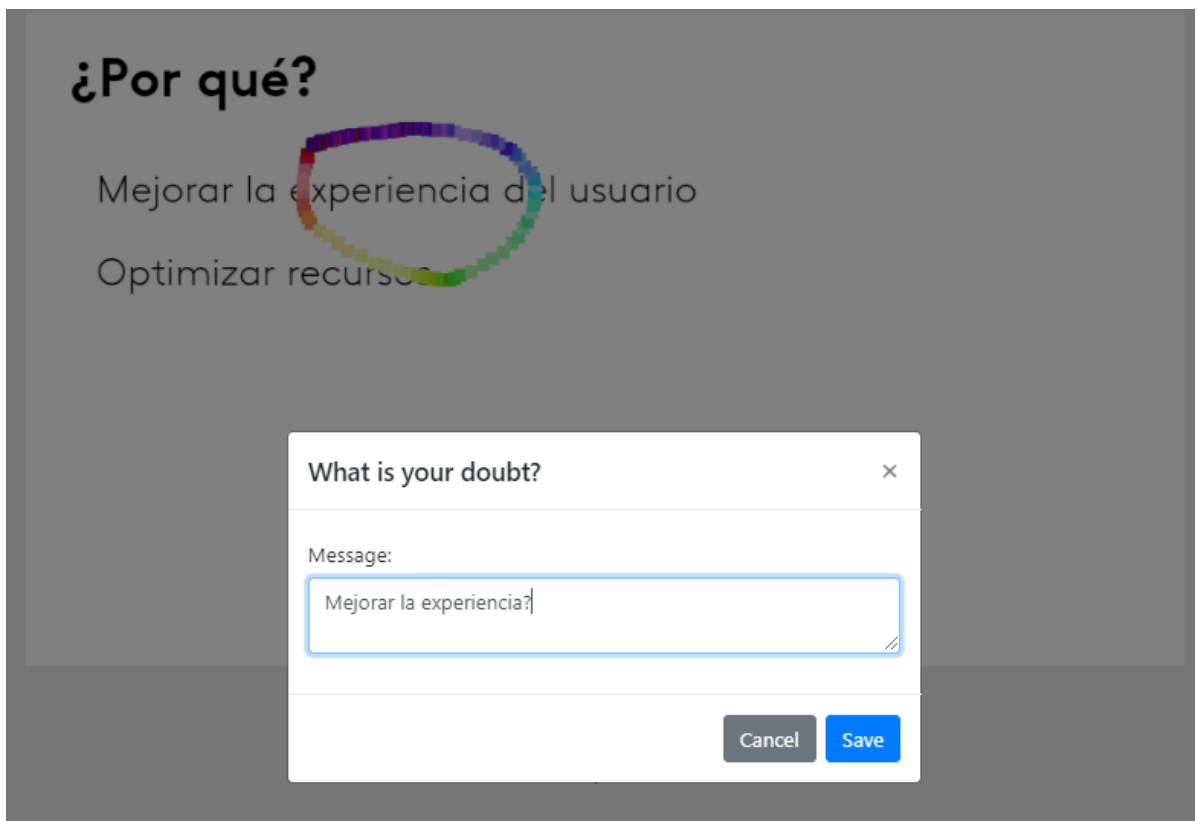


Figura 19: Recorte para generar dudas en la presentación

Para poner la presentación a pantalla completa, tendremos que apretar el botón de arriba a la derecha para desplegar la botonera y nos aparecerà la opción de “full screen” en azul.

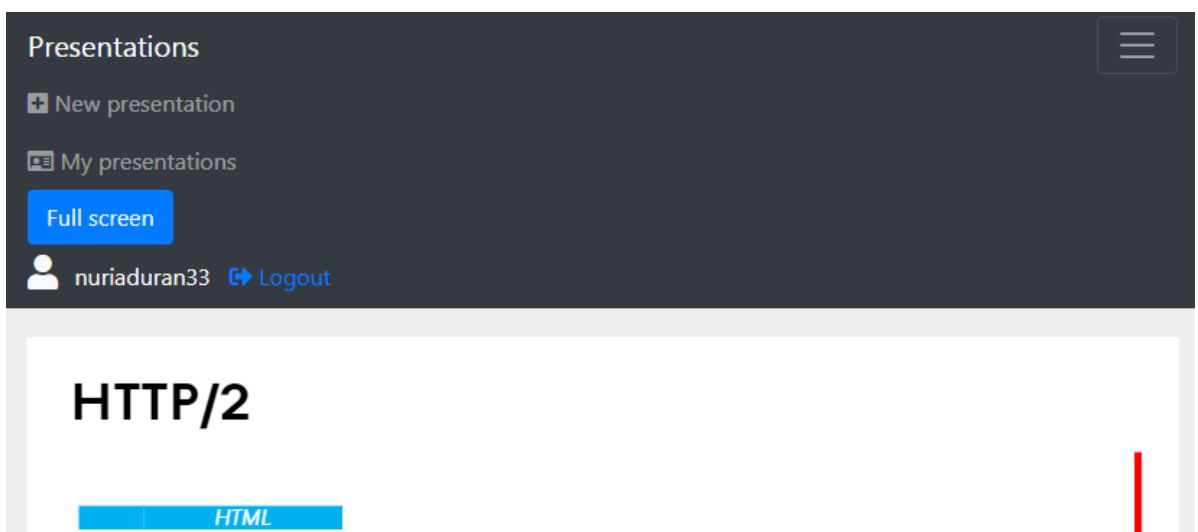


Figura 20: Recorte para poner la presentación en pantalla completa

Finalizada la presentación nos aparecerá una ventana con 3 botones: ‘Doubt reminder’, ‘Final results’ y ‘Rate presentation’. Cada uno de ellos hace aparecer un modal con distinto contenido. Si apretamos el botón ‘Doubt reminder’ aparece el modal que nos recuerda las dudas anotadas y en qué página, lo podemos ver en la siguiente figura 21:

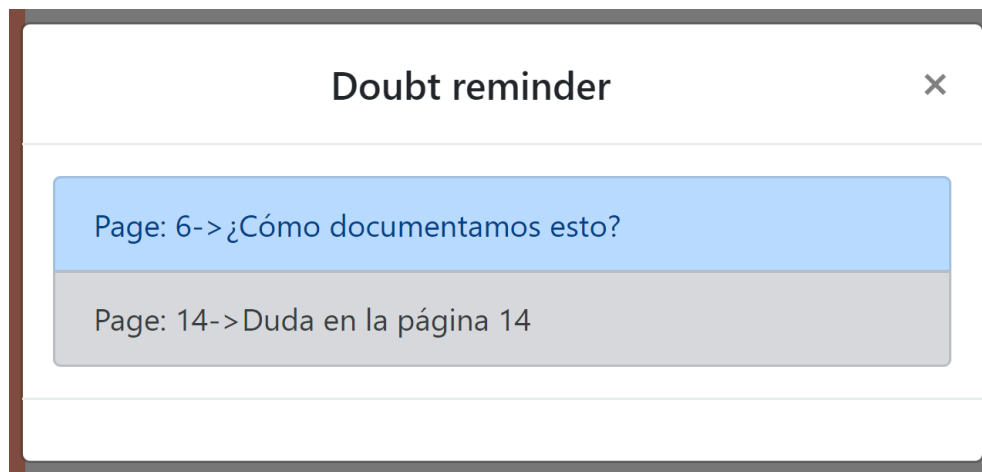


Figura 21. Recorte del modal de dudas

Si apretamos en el botón de “Final Results” podremos ver los resultados finales de las encuestas planteadas a lo largo de la presentación. El modal muestra un desplegable por cada encuesta. Podemos ver un ejemplo en la siguiente figura 22.

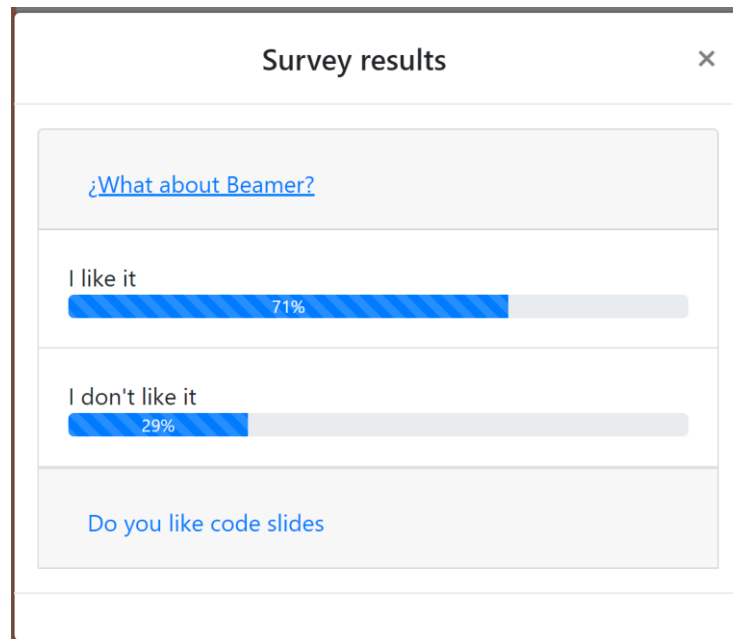


Figura 22: Recorte del modal de resultados de encuestas

Finalmente, si apretamos el botón "Rate presentation" podremos evaluar la presentación, siendo una estrella la puntuación mínima y 5 estrellas la máxima. Para puntuar hacemos click en las estrellas y para cambiar la puntuación volvemos a apretar a la estrella que corresponda. Enviamos la puntuación con el botón en azul "Submit". La siguiente figura 23 muestra el modal de puntuación.

The screenshot shows a modal titled "Thanks for rating :)" with a close button (X) in the top right corner. Inside the modal, there are several categories for rating, each followed by five stars (1-5 stars are highlighted in orange):

- Usefulness of the content presented
- Organization of the content
- Presentator skills
- Usefulness of the surveys
- General satisfaction

At the bottom of the modal, there are two buttons: "Cancel" and "Submit".

Figura 23: Recorte del modal de puntuación

En el caso de querer volver a la pàgina inicial apretaremos encima del nombre/título “presentations” situado en la parte superior izquierda de la web.